

SCULPT-ANIMATE

4D

Professional 3D Animation Software
for the Amiga Personal Computer



Created entirely with Sculpt-Animate 4D


BYTE by BYTE™
CORPORATION

FOREWORD

This manual and the software described in it are copyrighted with all rights reserved. Under the copyright laws, neither this manual nor the software may be copied, in whole or in part, without written consent of Byte by Byte Corporation. The software program may be copied for backup purposes, or installed on a single hard disk, or both. Multiple installations are expressly forbidden. Site license agreements are available.

In other words, you may (and should) make backup copies of the Sculpt Animate 4D distribution disks. You may also copy the program onto a hard disk for more efficient use. Please do not make copies of the program for others. You are honoring the copyright if only one copy of the software you purchased is being used at any one time.

When we released Sculpt Animate 4D's predecessors, we took a risk. We thought low-cost software could be sold without copy protection. We were wrong. It seems that some people who never see what goes into software development don't understand that a couple of pounds of paper and plastic can be a bargain at \$500. Yet this software compares with packages costing 100 times as much. Given the popular (extreme) falsehood that software developers are all rich, it is entirely too easy to give copies to friends, etc.

This has made continued development of Sculpt Animate extremely difficult. It has literally become a decision of whether to kill the series or to implement some protection against illegal copying. We will not kill the series! Instead, we have chosen a form of protection--the simplest and least hazardous we could find.

It is this simple: you may run any copy of the program, but when it starts you will be asked to type in a word from a specific place in the manual. If you do this correctly, the program will proceed normally. This scheme is as painless (for all concerned) and as effective as any other.

Limited Warranty on Media and Manual:

If you discover physical defects in the media on which this software is distributed, or in the manual distributed with the software, Byte by Byte Corporation will replace the media or manual at no charge to you, provided you return the item to be replaced with proof of purchase to Byte by Byte Corporation within 90 days of the purchase date. Please include your warranty registration card if you have not already sent it in.

Registered Sculpt Animate 4D owners may purchase a technical support policy at \$15 for ten minutes or \$75 per hour. Technical support is free of charge for 30 days after date of purchase. We **must** have your warranty registration card before any assistance can be given. Support is **only** available at:

(512) 343-4376 (10am-5pm, CST)

Please do not call for technical assistance until you are sure that this manual does not answer your question. The manual is very complete, and it can explain the program much more effectively than we can over the phone. It will also be easier for you to describe your difficulty if you are familiar with the manual's explanation.

Copyright (C) 1988 Byte by Byte Corporation
Arboretum Plaza II
9442 Capital of Texas Highway North, Suite 150
Austin, TX 78759

SCULPT ANIMATE 4D written by Dr. Eric Graham

Manual: Written by Cathryn E. F. Graham
and Marcus Brooks

Cover: Art by Rick Unland
Layout by Tony Tomsu and Steve Bittick

Printing: AlphaGraphics #151 and The Whitly Company

Revision 1a

CONTENTS

INTRODUCTION

What is Sculpt Animate?	xi
About this manual	xii
Notes for upgraded users	xv

CHAPTER 1: AN OVERVIEW

Sculpt Objects	1-1
The Sculpt Universe	1-3
Images	1-4
Animations	1-5

CHAPTER 2: GETTING STARTED

The menu	2-5
Loading an image	2-8
Loading a scene	2-11
Making a picture	2-18
Saving an image	2-20
Saving a scene	2-21

CHAPTER 3: THE OBSERVER

Imaging modes	3-1
Placing the observer	3-8
Aiming the observer	3-8
Camera lenses	3-9
Exposure control	3-10
The Jaggies	3-11
Dithering	3-12
Image size on the screen	3-12
Tilting the camera	3-14
Starting an image	3-15
Hiding and restoring an image	3-15
Aborting an image	3-16
Memory usage	3-17
Other status display items	3-19

CHAPTER 4: BUILDING OBJECTS

Vertices	4-1
Edges.....	4-2
Faces.....	4-3
Selecting vertices.....	4-4
Erasing objects.....	4-9
Color.....	4-10
Texture	4-13
Smoothing.....	4-15
The Grabber.....	4-17

CHAPTER 5: POWER TOOLS

Spheres.....	5-1
Subdividing objects.....	5-3
Other curved objects	5-4
Cubes.....	5-4
Prisms.....	5-6
Drawing curves	5-6
The Fill command.....	5-7
The Edge Maker Tool.....	5-8
Spinning new objects	5-9
Helices.....	5-10
Extruding shapes	5-14

CHAPTER 6: MORE TOOLS

Saving and loading objects.....	6-1
Rotating objects	6-3
Expansion and contraction.....	6-5
Finding the center of an object	6-7
Manipulating shapes.....	6-7
The magnet tools.....	6-8
The Selector tools	6-10
Duplicating Objects.....	6-11
Building mirror images	6-11
Multiple use of the extrude tool.....	6-12
The unslice operation	6-14
Hiding objects in the Tri-View.....	6-15
Creating complicated objects.....	6-16

CHAPTER 7: THE WORLD

Creating lamps.....	7-1
Illumination.....	7-4
The sky.....	7-5
The ground.....	7-6

CHAPTER 8: SPLINES

Splines and knots.....	8-1
Slope and speed.....	8-3
Cusps.....	8-7
Using splines	8-8
An example: A gearwheel.....	8-9

CHAPTER 9: WHEELS WITHIN WHEELS

Hierarchies of objects.....	9-1
Building a hierarchy.....	9-4
Using an object hierarchy	9-9
Editing the hierarchy	9-10
Loading and saving named objects.....	9-12

CHAPTER 10: MOTION PATHS

Paths.....	10-1
Creating paths.....	10-2
Tumbling.....	10-4
Motion hierarchies	10-11
Local Origins	10-14

CHAPTER 11: CREATING AN ANIMATION

The Take.....	11-1
Two kinds of animation	11-4
Global Animation.....	11-6
Example: global animation	11-6
Previewing an animation.....	11-9
Looking at individual frames.....	11-11
Key frame animation.....	11-12
In-betweening	11-14
In-betweening distortion	11-17
Example: key frame animation.....	11-18
Global vs. Key frame.....	11-20

RAM animation	11-22
Making a RAM animation.....	11-24
Playing a RAM animation	11-26

CHAPTER 12: ADVANCED TECHNIQUES

Loop mode	12-1
Frame duration	12-2
File name conventions.....	12-4
Disk space.....	12-5
Organizing floppy disks.....	12-6
Saving images.....	12-8
Color and exposure locking.....	12-8
Animating the observer.....	12-10
Animating lamps	12-11
Model Animation.....	12-12
Object Cycling	12-14
Animated objects.....	12-15
Backgrounds and foregrounds.....	12-16
Motion blurring	12-18
Single frame controllers.....	12-18
Frame buffers	12-19
Connecting up the hardware	12-20
Using a GenLock	12-21
Genlocking the Tri-View	12-22

CHAPTER 13: BY THE NUMBERS

Sculpt Animate coordinates.....	13-1
The Coordinates window.....	13-2
The coordinate origin.....	13-4
The tape measure.....	13-5
The protractor.....	13-6
The grid	13-7

CHAPTER 14: KEYSTROKES, FONTS, SCRIPTS

Keyboard menu equivalents.....	14-1
User settable hot keys.....	14-2
FonTS	14-3
Scripts	14-6
Script commands	14-8

Batch mode.....	14-20
Conclusion.....	14-21

APPENDIX A: FUNCTIONAL REFERENCE

The Tri-View Gadgets.....	A-2
Menu commands.....	A-6
The file requester.....	A-32
The hierarchy requester.....	A-33
The expansion requester.....	A-34
Color requesters.....	A-35
The modify faces requester.....	A-35
The modify lamps requester.....	A-35
The modify knot requester.....	A-36
The modify tumble requester.....	A-37
The modify Take requester.....	A-38

APPENDIX B: GLOSSARY

APPENDIX C: FILE FORMATS

APPENDIX D: ERROR MESSAGES

INDEX



PREFACE

Several years ago one of my professors gave an interesting definition of an animator.

"When you find someone who's good at animation," he said, "you will also have found an extremely weird bird."

His point was that animation required infinite patience, good math skills, artistic talent and technical savvy. Those of us who lacked these attributes were put on notice to turn our efforts to the more generalized techniques of film and video production. And leave animation to the *weird birds* of the world.

I don't know that Dr. Eric Graham ever heard this definition. Thanks to his Sculpt-Animate series of programs, however, anyone can produce an animation these days. Weird bird or not.

The Sculpt-Animate series is one of the most amazing graphics packages to reach the personal computer. The menus and methods are straightforward. Logical. Beneath this interface, then, lurks powerful 3D code--capable of operations that were once in the exclusive domain of mainframes.

Given this easy interface, someone who has some knowledge of 3D representations and/or animation might proceed to animating simple objects shortly after boot-up. For the rest of us, a few observations may be in order.

As you read this manual, look at the objects around you. Don't just see what they look like, but how they are shaped. You can see that almost every object has clear and definite features, outlines, and dimensions. Every feature has a certain position with respect to the rest of the object, and the world. These are the elements used to model objects in Sculpt Animate.

INTRODUCTION

What is Sculpt Animate?

Sculpt Animate 4D, the latest member of the Sculpt-Animate series, is an integrated solid modeling, animation, and image rendering system. The program enables you to design three-dimensional (3D) objects and display them as still or moving images, in full color.

This means that you can use Sculpt Animate to construct a computer simulation of virtually any 3D shape that your mind can conceive, and then see what it would look like by displaying its image on the computer screen. You don't have to be a mathematician or know anything about 3D geometry to use Sculpt Animate.

Any shape can be formed and manipulated with the aid of the mouse and its two buttons. You are not limited to simple geometric shapes like spheres and cylinders. Indeed, with a little imagination you can build 3D models of almost any real-world object. With some artistic skill you can even create very difficult irregular shapes, such as human and animal forms.

Regard Sculpt Animate as the next step beyond a paint program. A paint program enables you to daub simulated pigments on the screen of your display without having to deal with the physical encumbrances of painting, and without messing up your best shirt. Using a paint program effectively requires artistic skill, however, particularly if you are trying to produce a naturalistic scene.

Sculpt Animate takes care of many of the messy technical details that any 2D artist would have to deal with, such as perspective, shading, and shadows. In fact, making an image with Sculpt Animate is very much like taking a photograph, just point and shoot and out comes a picture.

When you are using Sculpt Animate to build a solid object, the construction is much easier than with any real construction medium. Changes can be made easily, and you can save your intermediate stages. If you make a mistake or change your mind you can quickly backtrack.

Sculpt Animate is not limited to making still images. The same tools that you use to build three dimensional objects can also be used to describe motion paths for those objects, or to specify the way an object is to change its shape or position. Sculpt Animate can then be instructed to make a series of images showing the motion that has been described.

The data for some moving images can be compressed enough so that the animation can be displayed directly by the Amiga, this technique is called RAM-Animation and was used for the 'Juggler' and 'Kahnankas' demonstrations, as well as others that have been shown throughout the Amiga community. With appropriate hardware and drivers, an animation that's too ambitious for RAM playback can be generated for storage on film or video tape.

Software packages like Sculpt Animate have been available for some time. What makes Sculpt Animate unique, besides its extra-simple interface, is that the earlier packages cost about one hundred times as much and required hardware that cost a hundred times more than an Amiga.

What is the catch? Well, Sculpt Animate can only display an image within the constraints of the Amiga hardware. It can never generate a big-screen image as detailed as those seen in the movie 'Tron'. Similarly, because the Amiga is not a supercomputer, some tasks that Sculpt Animate can perform will take a while. Does your Amiga really work 24 hours a day? Now it can be doing useful work while you are asleep, or away at the office!

Seriously, while it could take many hours to make an image of a taxing scene in the best available quality, excellent images can be created without going to such extremes.

After you are comfortable creating object models, sit back for an hour or so and watch a few animations. Observe how the characters really move. Study how limbs operate, how objects rotate and change shape. See how your eye blends many discrete images together, perceiving smooth motion. These are the elements used to create animations in Sculpt Animate.

Read this manual once to acquaint yourself with the general method. This first reading is a nice time to link the Sculpt Animate strategy with your own observations regarding objects and animation. By the second reading, you'll be like a kid on Christmas morning. As you do the tutorials, you'll begin to get the feeling that Dr. Graham has given you a *weird bird* all your very own. Your objects will be made to fly, bank, walk, and squash. In so doing, Sculpt Animate will open up an exceedingly powerful medium limited only by the boundaries of your imagination.

Ten years hence, when rendering is done in real time, disk storage is tabulated in gigabytes, and the average RAM-count is the size of a Buick, Sculpt Animate will still be around, lighting the way. Given these anticipated hardware advances, the Sculpt Animate sequences of the future may be indistinguishable from camera-reality. And that brings us to another definition. My old college professor also reminded us that it doesn't take genius to make the complex... complex. "It takes genius to make the complex simple." By that definition, the Sculpt Animate series is a work of genius. Sculpt Animate flies fast and high. It also flies low enough so that non-computer/non-math types like me, guys that can't even spell *cowkulus*, much less do it, can jump onboard and do the animations we've always dreamed of.

Floyd Wray
Earth & Sky Films

INTRODUCTION

Chapter 1 provides a grand overview of the program, to whet your appetite and to give you a feeling for where you are and where you are going when you move on to the chapters that follow.

Chapter 2 introduces you to the use of Sculpt Animate's solid modeling and static image generating capabilities, and lets you familiarize yourself with the essential elements of operating the program. A section at the end of this chapter provides troubleshooting hints.

Chapters 3 through 9 thoroughly explore the static imaging and modeling aspects of the program, progressing from the basics through advanced techniques, and laying a foundation for animation.

Chapter 10 introduces Motion Paths, the first element of the program which is entirely unique to the purpose of creating animations. Don't think that you can easily skip to this chapter, though. Every aspect of Animate is built firmly upon Sculpt's foundation.

Chapters 11 and 12 complete the presentation of animation techniques. Chapter 12, in particular, details special cases and advanced techniques which may be used.

Chapter 13 explains coordinates, grids, and other positioning aids which can be used throughout Sculpt Animate to fulfill specialized purposes.

Chapter 14 details the use of keyboard equivalents, hot keys, fonts, scripts, and other 'power user' methods for accelerating your work.

A reference section follows the main text. This contains appendices which may be consulted for concise definitions of Sculpt Animate functions and terminology.

Finally, an Index is provided after the reference section.

Sculpt Animate also provides ways to get a quick preview of what the final image will look like, in only a minute or two.

Even within its hardware-imposed constraints, Sculpt Animate offers you a powerful ability--the ability to build your own private universe. Once you have finished construction, you can roam your imaginary world and take pictures from any viewpoint, in any direction. Look at ordinary objects in extraordinary, otherwise impossible ways, or look at objects that have never existed, and see what they'd look like if they did.

Indeed, while we have been emphasizing Sculpt Animate as a recreational and artistic tool, it is by no means limited to this role. Professionals involved in construction or manufacturing--of buildings, vehicles, coffee mugs, or whatever--can use Sculpt Animate to view designs before committing to actual production. Scientists can present or visualize objects that couldn't be seen otherwise, such as molecular shapes or mathematical constructs. Videographers can create flying logos and moving illustrations to give their work visual impact.

No special skills are needed to operate Sculpt Animate. You can 'take a picture' at any time, to see if your 3-D objects are shown the way you'd like them to be. This manual will lead you through all the stages of solid modeling and image generation. Unlike many programs, you can start getting useful results from Sculpt Animate with a minimum of training. In a few pages you will be told how to generate your first picture, and by the time you have finished the book, there will be no scene that you cannot create.

About this manual

The main portion of this manual is organized as an introductory text, which walks you through Sculpt Animate feature by feature, explaining what each feature does and how it may be used in the creation of your objects, images and animations.

Notes for upgraded users

This version of Sculpt Animate naturally differs from earlier versions. This section is not intended to enumerate all of the changes and additions that have been made. Some of the changes made, however, although always made with the intent of increasing the program's efficiency, might prove momentarily confusing to users who are familiar with earlier versions.

The most disrupting changes take the form of menu shuffling. The greatest of these is that all of the face attribute functions--color, texture, and smoothing--have been combined into a single requester, EDIT MODIFY FACES. This has eliminated several menu items, while preserving all of their functionality.

A similar change has been made to the Lamp functions. EDIT LAMPS COLOR and EDIT LAMPS BRIGHTNESS have been combined as EDIT MODIFY LAMPS. LAMPS CREATE has been changed to EDIT ADD LAMP.

All of the LOCAL ORIGIN commands have also been moved to the EDIT MODIFY submenu, combined in the EDIT MODIFY LOCAL ORIGIN requester.

The PROJECT ANIMATION submenu has been broken up, and its functions moved into the PROJECT LOAD, PROJECT SHOW, and PROJECT UNLOAD submenus.

Two other moves have been made for purely logical reasons. EDIT DO BE SPHERE has become EDIT SNAP CONNECTED TO SPHERE, and OBSERVER SHOW has become PROJECT SHOW IMAGE.

These are by no means the only changes that have been made to the program. The many other changes are primarily additive, however, being features that simply never existed before. Other changes should be obvious enough that you won't have to think twice about them.

INTRODUCTION

This manual and those of earlier Sculpt Animate products naturally have several similarities. Be sure to read through this manual at least once, though. It thoroughly explains the program's new features, and great effort has been expended to enhance and clarify the rest of the text as well.

CHAPTER 1

AN OVERVIEW

This chapter contains an overview of Sculpt Animate. Everything in this chapter is described elsewhere, so you may skip ahead if you wish. You may find it useful to read what follows, though, so that you have a grasp of the big picture before we delve into all the details of the program in the other chapters.

While Sculpt Animate is easy to use, some aspects of its use are not immediately obvious. You may want to begin using the program immediately. This is a good way to get your feet wet, so to speak, but you can quickly get in over your head. If this happens, don't be too discouraged, just turn back to this manual and continue reading.

You will find that there are several different ways to do certain things. If what you are trying to do seems too difficult, you can probably find an easier way if you review the related portions of this manual. When a reference is made to another chapter, be sure to look there as well.

Sculpt Objects

The objects that we will be building, be they cubes, aircraft or elephants, will all be made from a single building block--a triangular face. Such faces can be joined together to approximate the surface of any shape. The smaller and more numerous the triangles, the better the approximation. This needn't be carried very far, however, for although the triangles would normally create a faceted, angular shape, they can be shaded as if they were curved. This makes the joint between adjacent triangles quite invisible. This is called Phong smoothing, and it's particularly easy to employ, you'll just set smoothing 'on' for certain faces, and the program will take care of the details.

Each triangular face can be given other attributes besides smoothing, such as color and surface finish. Faces can be shiny or dull, luminous or metallic, even mirrored or semi-transparent, so you will be able to accurately model a wide range of materials.

Although the triangular face is the basic building block for objects, each face is described in terms of simpler objects called vertices and edges. A vertex describes a position in space representing the corner of a face, while an edge is a link between vertices, and is used as the side of a face. Vertices and edges can be shared between more than one face, so Sculpt Animate objects are really built up from a triangular network of edges. By moving one or more vertices, you can change the shape of an object.

Objects are built in an interactive environment called the Tri-View that displays the vertices and edges in a so-called wire frame view. The Tri-View lets you look at a box-like region of space from three different angles at once, usually northward, westward and downward. You can use the mouse to manipulate vertices in all three directions, but you will have to switch from one view to another to be sure that you have created the intended change.

You need to use multiple views because a window on a flat screen can only represent two dimensions at once. Think of a shoebox: if you look directly at its end, you can only see two of its dimensions--height and width. If you want to look at its length, you have to look at either its top or side. The Tri-View allows you to do this. Although it may seem difficult at first, especially if you are new to such things, take heart. After a little practice you will be surprised by the intricate structures you can build.

Sculpt Animate gives you commands for producing many standard shapes automatically. These can then be stretched or distorted to produce still more shapes. Shapes that have an element of symmetry can often be generated from simple curves, for example a wine bottle shape can be generated by building its profile and using the Spin command to sweep the profile around, creating a curved

surface. Similarly, the Extrude Tool will let you stamp out shapes like a cookie cutter, and the Unslice command will join cross-section profiles, creating quite irregular shapes.

Sophisticated users can create spline curves: smooth curves made of many vertices, of which only a few need be adjusted to alter the curve's shape. Objects can also be decomposed into named hierarchies, making it easier to manipulate parts of an object while leaving other parts unaltered.

Objects can be saved and recalled from disk, so you can quickly assemble a library of useful objects. Once built, these can be reused in any number of scenes. Because of the success of programs in the Sculpt Animate series, there are numerous objects whose builders have placed in the public domain, or included in commercially available object libraries. When you use pre-constructed objects, it is almost trivial to assemble interesting and complicated scenes. Dazzling images can be created with an absolute minimum of effort.

The Sculpt Universe

Once you have assembled your objects, Sculpt Animate can be asked to provide certain environmental elements to round out the scene. One useful item is a ground plane that stretches off to infinity. The ground can be given any solid color, or painted with a checkerboard of two contrasting colors. A sky can also be provided which can be shaded from one color at the zenith (overhead) to another color at the horizon. If your demands for a background setting are more complicated--for example if you need a distant mountain range, or a lake of rippling water--then these can be created in the form of objects.

A major purpose of Sculpt Animate is to create images of your modeled objects, and the most important element of any image is its point of view. In the Sculpt universe, this is called the 'Observer'. Think of the Observer as a camera that can be positioned and pointed anywhere. The view that

OVERVIEW

this hypothetical camera would capture is what your screen displays when an image is rendered. You are provided with a number of controls over the Observer, including a lens with an alterable focal length, so that you can zoom in or out of the scene and modify the apparent perspective of the image.

In addition to objects and the Observer, a scene will normally contain lamps to illuminate the objects. As any painter, theatrical set designer, or photographer knows, the placement of light sources can be instrumental in creating a desired effect. In Sculpt Animate, any number of lamps can be used, and each can be given a different position, color, and brightness.

Images

There are numerous ways to produce a computer generated image. As you might expect, the better quality images usually require more computer time. While simple images can be produced in a few seconds, realistic images with shadows, highlights and reflections can take hours or even days to generate. Sculpt Animate allows you to trade image size and realism for savings in computing time. Numerous rendering modes and image sizes are provided, so you can pick the degree of realism you require for a particular application. For complex scenes, it's best to try one of the faster modes or a smaller size to preview the scene before committing your Amiga to a lengthy computation.

Images can be stored on disk in the Amiga's standard IFF format. Images can also be recalled from disk to be displayed on the screen. The images can be modified using either paint programs or image processing programs, although not every program can handle the Hold-and-Modify (HAM) or the overscan images that Sculpt Animate is capable of generating.

Sculpt Animate can even generate images that an Amiga cannot display. These exist as raw data representing almost any desired resolution and up to sixteen million colors. The

data can be sent out to files for programmers and the like to use, or to a separate frame buffer, which is a device built just to display high quality images. Each frame buffer type is accessed through its own custom device driver, so Sculpt Animate won't have to be re-written for every new frame buffer that appears.

Animations

Sculpt Animate can be used to make animations. Objects can be made to move by specifying their positions at different times. Each feature of the Sculpt universe can be made to change, lamps can dim or brighten and change color, objects can change size and shape, or even vanish, the observer can fly through the scene and change its viewing direction, the sky can be made to darken to simulate dusk. There is practically no limit to the effects that can be animated.

An animation comprises many images, each differing from the previous one by just the right amount to create the illusion of smooth motion. Sculpt Animate uses a mechanism called a 'Take' to keep track of all the details of a single animated sequence. The Take knows the names of all the files that describe the objects as they move around, as well as the name of each image that is generated. When a Take has been properly specified, the process of rendering all the frames can be accomplished with a single mouse click.

Object movement can be specified in several ways. You could, for instance, create a series of scenes, setting the position of each object for every picture in the animation. This would be extremely tedious. Since most motion is smooth, however, it's much easier just to set up a few images called key frames. The program can usually figure out where the objects should be for the frames in between. This technique is called key frame animation.

If an object's motion is fairly simple, for example a car driving along a road, the path that the object follows can be

described, and the object 'attached' to the path. The path describing the motion is just another object, so a single scene can contain all the information needed to animate the object. This is called global animation. Very complicated motions can be described by a hierarchy of motions, with movements superimposed on each other to produce complex motions. Global and key frame animation can even be mixed in the same take, with some objects being described in a global manner while others are described in terms of key frames.

The motions can represent actions that are impossible in the real world, or they can simulate reality with uncanny accuracy, it is up to you to decide what effect you wish to create.

There are two ways for completed animations to be displayed. If the animation is not too large, is possible to store its images in memory in compressed form, then display them quickly enough to simulate smooth motion. At video frame rates, however, even a second's worth of animation can use a lot of memory, and if motion is extreme, the time required to reconstruct each image may prevent full speed playback. Nevertheless, RAM animations can be useful demonstrations, or can be edited together onto videotape for longer presentations.

The other, more versatile way to display animations is to copy the images directly to motion picture film or to videotape. As you might expect, this method requires the use of special purpose hardware. Such devices require custom 'interface' software. Sculpt Animate can access this software through a device driver facility, without any need to customize Sculpt Animate itself.

As you can see, Sculpt Animate offers a great many facilities, which you can use to create sophisticated images and animations. Sophistication is not a prerequisite for the user, however. While it is true that advanced tools and 'power user' techniques exist, the essence of Sculpt Animate is to open up the world of 3D graphics and animation to everyone. Here it is, climb in!

CHAPTER 2

GETTING STARTED

This chapter introduces you to the basics of operating Sculpt Animate. You will be shown how to load a scene from a disk, generate a picture of the scene, and finally save the picture to disk.

Before you do anything else, read the support policy statement and warranty information that came with this package. Fill out the warranty registration form and send it in. This ensures that you will be able to obtain support when you need it, and registers you for future updates and announcements.

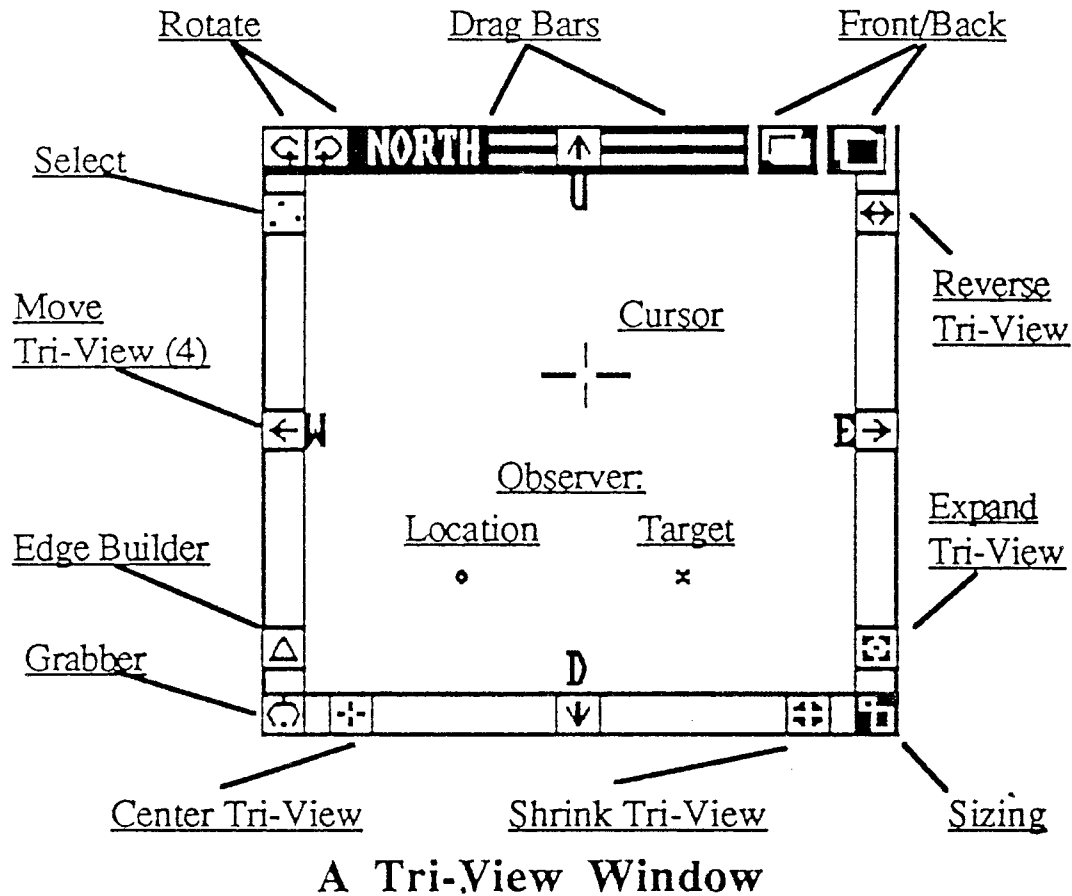
If you have already used a few Amiga software packages you can probably skip to the next section, but be sure to make a backup copy of the Sculpt Animate distribution disks first. If you are new to the Amiga, you should know that while floppy disks are very secure, accidents can happen, so it is wise to make a copy and keep the original disks in a safe place, far from coffee spills, magnets, and small animals.

The book Introduction to the Commodore Amiga, that came with your Amiga, tells you how to make backup copies. When this important step is complete, you are ready to start using Sculpt Animate.

The program can be started from the Workbench Screen by first double-clicking on the icon for the Sculpt Animate disk. Then, when the disk's window opens, double-click on the 'SA' icon that appears. If you are an advanced user you may run SA from the Command Line Interface (CLI), but first you must set the CLI's stack to 50,000 or more.

After a few moments of disk activity, Sculpt Animate starts by displaying three windows, arranged like three sides of a folded-out box on the screen. These are collectively known as the Tri-View.

NOTE: If your system disk (usually 'Workbench') is not in a drive, it is likely that a requester will appear when you run Sculpt Animate, asking you to insert the desired volume. This means the program needs to read a file on that disk, and can't proceed without it. Whenever this happens, do not click CANCEL. Instead, insert the requested disk and click 'Retry.'

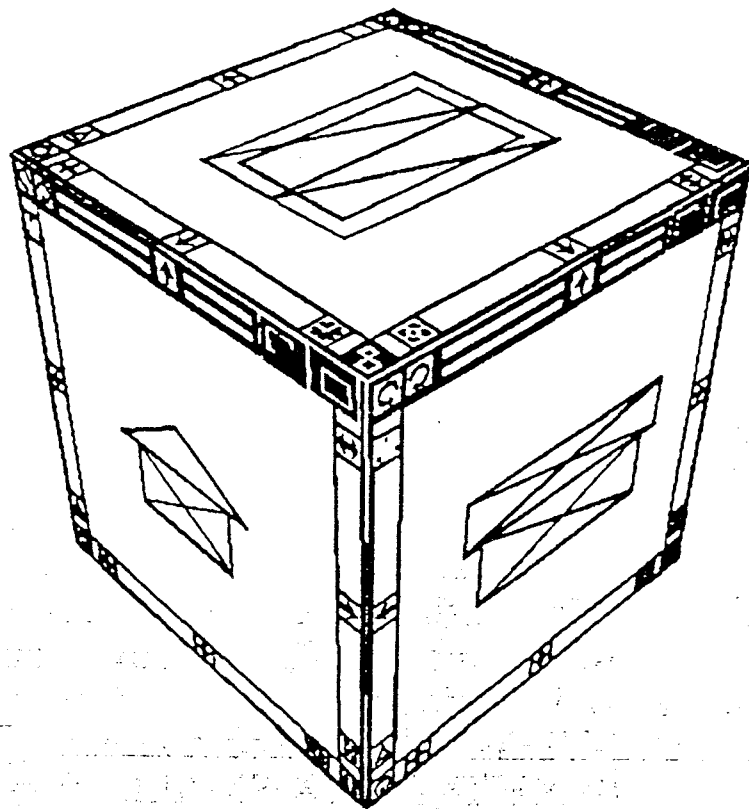


The Tri-View

The Tri-View is a necessary part of three dimensional editing, because the Amiga can only display flat images on the screen, while Sculpt Animate works with three dimensional solid objects. The three windows show three different views of a rectangular chunk of space. Initially each window is empty except for a little blue cross in the center, and four letters denoting the cardinal directions that apply in that window.

The illustration shows a representative Tri-View window, there are three windows like this on the Sculpt Animate screen. The window at the upper left is a view looking downward, like a map, with north at its top, west to the left, etc. The window at the lower left shows a view looking northward, so west is to the left and up is towards the top of the screen. The third window, at the lower right of the screen, is a view looking west, with south to the left and north to the right of the window.

If you have ever encountered technical drawings, or blueprints of a house, you will be familiar with this way of representing solid objects. As with most Amiga software, these windows may be moved around on the screen in any way that you wish. To move a window, use the mouse to position the pointer anywhere on the Drag Bar at the top of the window. If you press and hold the left mouse button on this bar, an outline of the window will appear. This outline can be moved anywhere on the screen. When you release the button, the window will be re-drawn at the new position.



The Tri-View "Box" in 3D.

The size of a window can be changed by using the sizing gadget that can be found at the bottom right corner of each window. To change the size of a window, use the mouse to point at the sizing gadget, then hold down the left mouse button while you drag the corner of the window to create a new shape or size. When you release the button, the window is refreshed, but you may notice that something funny happens. Whenever you change the size of one window, one or both of the other windows will change also!

This is because the windows represent a rectangular chunk of empty space, they're like three sides of a box that has been opened out and laid flat. If only one of the sides is changed, then it won't fit when the box is folded back up. One or both of the other sides will have to be changed as well, so the program does this automatically. Suppose that you reduced the size of the box at its south end, then the change would also be seen in both the view looking down and the view looking west.

If your windows overlap one another on the screen, you can use the back gadget or the front gadget to change their depth arrangement. As usual on the Amiga, these gadgets may be found at the right hand end of each window's top border. To use a gadget, point to it with the mouse and click the left mouse button.

The initial sizes of the Tri-View windows are small enough that each window can be displayed without covering another. This is recommended when you first use the program, because it is easy to get confused while working in three dimensions.

When you have gained confidence, you may wish to make the windows as large as possible and use only one window at a time, using the depth arrangement gadgets to go from window to window. The windows can be re-sized manually, but it's quicker to use the command `EDIT DO MAKE TRI-VIEW BIG`. The complimentary command `EDIT DO MAKE TRI-VIEW SMALL` will return the windows to their default configuration.

Each Tri-View window also has more than a dozen extra gadgets tucked away in its borders. We will be describing each one in detail later in this book. To avoid confusion, try to restrain yourself from using the gadgets, until you know what they are supposed to do. Of course, if you feel adventurous you can try them anyway. None of them can cause permanent damage to your Amiga.

If you use a gadget incorrectly, a message may appear to warn you of this. Such messages usually display a 'button' gadget, marked 'CONTINUE', beneath their text. To continue with the program, move the mouse pointer into the button's outline and click the left mouse button.

The blue cross shown in each Tri-View window is called the cursor. There is only one cursor, but you see it in all three windows at once. It represents a unique position in three dimensional space. To move the cursor, point with the mouse at a new position in any window and click the left mouse button. The cursor will jump to the pointer's position. If you hold down the left mouse button and move the mouse, the pointer will vanish and the cursor will move in its place.

You will notice that you cannot move the cursor outside any window. You should also notice that when you move the cursor in one window, it also moves in the other windows. The mouse can only move in two dimensions, but the Tri-View makes all three dimensions readily accessible anyway, just by switching to another window. You will quickly become accustomed to this style of working in 3D.

The menu

Most of the commands that you can issue to Sculpt Animate are made through the Amiga's menu system.

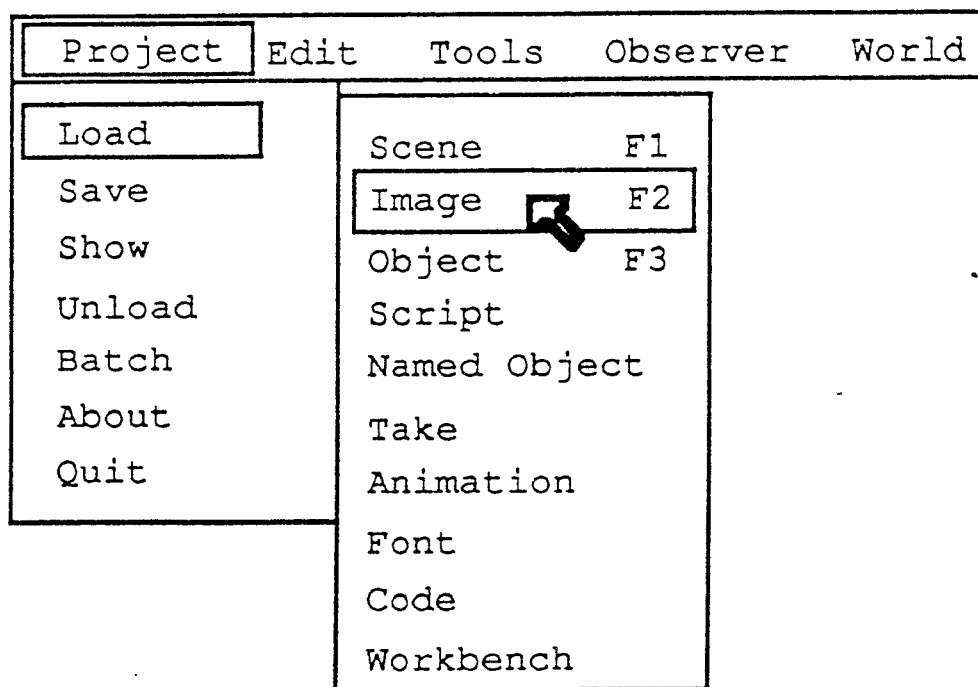
To make the menus visible, press the right mouse button and hold it down. If nothing happens, move the pointer into one of the Tri-View windows, and click the left mouse

GETTING STARTED

button (this 'selects' the Tri-View). Then try again with the right mouse button.

The menus show up as a bar at the top of the screen, showing the words 'PROJECT', 'EDIT', 'TOOLS', 'OBSERVER' and 'WORLD'. These are the major menu headings. To see the choices in each menu, move the pointer up to the desired heading, keeping the right mouse button down all the while. When the pointer touches one of the words, a list of choices will spring down.

To activate one of these choices, move the pointer to that item and release the right button. If, as you look through the menu, you decide you'd rather not execute any of the commands, move the pointer away from the menu region and release the right mouse button. Be a little cautious about where and when you release the button, or you may execute a command unintentionally.



Menu and Submenu

As you move the pointer down the list of menu choices, sometimes an additional set of choices will pop up to the right. This is a 'submenu.' To execute a submenu item, keep the right mouse button pressed and move the pointer to the right, then down the new list to the item you want.

When you release the button over an item, that action takes place.

At times you may need to issue several commands from the menu, one after another. Rather than releasing the right button, and then having to wend your way throughout the menu system each time, there is a way to generate multiple commands.

When the pointer is over the menu choice that you want, you can keep the right button depressed and click the left button instead. Still holding down the right button, you can keep moving through the menus, selecting additional commands by clicking the left button on their entries. When you release the right mouse button, the items you selected will be executed in the order in which they were selected.

There is one exception to this, if one of the items selected brings up a requester (requesters are discussed below), subsequent selections in a multiple command will be ignored. You will quickly become proficient at this speedy way of using the menu system.

For those of you who are not averse to dealing with the Amiga's keyboard, there is an even more fluid and efficient way to get to the menu's commands. In the menu, you will notice odd two-character symbols to the right of some menu items. Each of these is a 'keyboard equivalent' for that menu item. If you are familiar with keyboard equivalents in other Amiga programs, you can use them now. If not, they are explained fully in chapter 14.

Keyboard equivalents can also be defined 'on the fly'. The simplest of these is the ESC key, which will repeat the last menu command. You can also hold down the ALT key and press any alphabetic key after executing a menu command. This defines that alphabetic key as a 'hot key' equivalent for that menu command for the rest of the session, or until the key is re-defined. This is also explained in chapter 14.

Loading an image

To get an idea of the capabilities of Sculpt Animate, take a look at some of the images it has created. Hold down the right mouse button and point to the word PROJECT at the top left of the screen. Move down to the word LOAD, and then move to the right to the word SCENE. Finally move down to the word IMAGE and release the right button. You have issued the command PROJECT LOAD IMAGE.

If you have done this correctly, the menus will vanish and the pointer will be replaced by an image of grinding gear wheels. This indicates that Sculpt Animate is busy and should not be disturbed. In this case it is looking through the disk to see if it can find any files that contain images. You will notice that the disk drive light is turned on.

After a few seconds, the normal pointer will return, and a new window will be displayed. This window is a 'requester'--the Amiga's standard method of asking the user for more input. There are many kinds of requester, this one is asking you to choose the name of an existing file that contains a Sculpt Animate image. This File Requester is used by several commands, so you will be seeing it a lot.

The File Requester contains three large boxes for making selections. The contents of the boxes will depend upon what files you have in your computer.

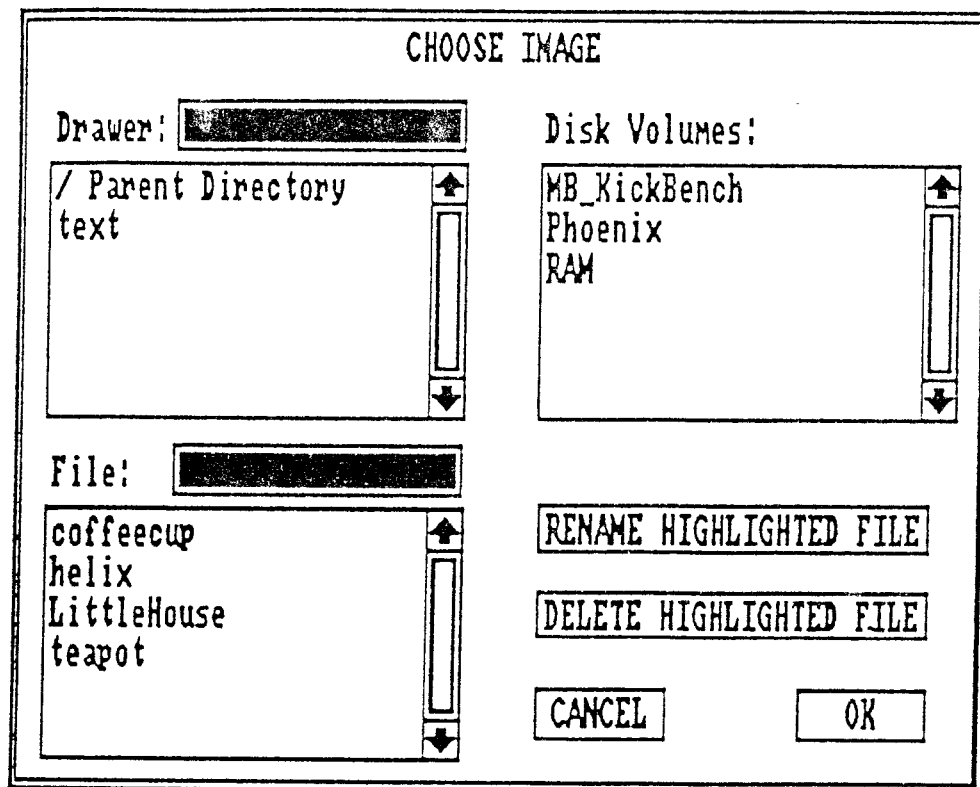
The box on the upper right is used for selecting a volume. Each floppy disk that the Amiga knows about is regarded as a separate volume, with its own name. The RAM disk is also a volume, as are any hard disks, etc., that may be mounted (recognized by the system). A volume is distinct from a disk drive or unit, in that a drive may exist that has no volume in it. Data resides in a specific volume, but the drive that a volume is mounted in is of little importance to Sculpt Animate.

If you wish to inspect the contents of a particular volume, double click on the name of the volume. After some disk activity, the display will change, and some of the

GETTING STARTED

contents of that volume will now be visible in the requester's other two large boxes.

The large box at the upper left of the requester contains a list of drawers, if there are any in the selected volume. You can double click on the name of a drawer to select it. The uppermost entry in this box is called '/Parent Directory' and it is used to select the drawer, if any, that contains the current drawer.



The File Requester

The large box at the lower left contains a list of the names of all the images in the selected drawer on the selected volume. At this time you could double click on the name of the file and the image would be displayed.

Above the Drawer and File boxes are slots where you can type in a name. At the lower right are four named boxes called 'RENAME HIGHLIGHTED FILE', 'DELETE HIGHLIGHTED FILE', 'CANCEL' and 'OK'. These boxes are called buttons--each one can be activated by clicking on it with the left mouse button.

The CANCEL button is present on most requesters and enables you to cancel the current command. The OK button informs the system to proceed with the current selection.

The RENAME and DELETE buttons can be used to change the name of a file and to delete it from the system, respectively. CAUTION: If you delete a file, there is no reprieve, its contents will be lost forever.

If there are more than six items to be shown in any of the three large listing boxes, only six items will be shown, but the gadget to the right can be used to reveal more. This gadget is called a scroll bar, and it is similar to scroll bars in other Amiga programs you may have used. The white rectangle in the tall narrow box is called the slider knob. The size of the knob relative to the size of the slider box indicates how many items are shown relative to how many exist.

You can grab a slider knob by placing the pointer over it and holding down the left mouse button. If you move the slider to the bottom of its box and release the left mouse button, the listing box will display the last six items in its list. If the knob is moved to the top, the first six items will be shown. With the knob in the middle of the slider box, the middle six names are displayed.

To move up or down in the displayed list of files, click on the arrow gadgets above or below the slider box. If the listing box contains six or fewer items, the slider will be inactive, filling its box to show that there are no extra items to display.

Advanced users, especially those with hard disks, may wish to keep different projects in different drawers. This makes it easier to find files and to keep things organized.

When you have found a file that looks interesting, double click on it. The screen will clear and the image will appear one line at a time, starting at the top of the screen. When you wish to stop viewing the picture, click on it first with the left mouse button, and then click with the right mouse button. If you want to remove the image from

memory, hold down the right mouse button, go to the PROJECT menu, move down to UNLOAD, then across to IMAGE and release the button.

Loading a scene

A scene is a collection of objects and lamps. A scene may also contain information about the observer and a description of the ground and sky. The scene is essentially a description of what Sculpt Animate is to make an image of, and how the image is to be made. Scenes are manipulated in the Tri-View and may be stored on disk. Your Sculpt Animate disk comes with some scenes, so let us try loading one of them.

Execute the menu command PROJECT LOAD SCENE. After a short pause, a requester will pop up that looks very much like the requester used to load images. Scroll the list until you find a scene called "LITTLEHOUSE". Double click on it to indicate that you have made your choice, and that you still want to load a scene.

Objects and lamps that are already in the Tri-View will not be erased when you load in a scene, but at this point a second requester will appear, giving you a choice of what parts of the new scene to load. You may selectively load parts of several scenes to combine them. This is explained more fully in chapter 6.

If you want to start with a clean slate, use the command EDIT ERASE ALL before using PROJECT LOAD SCENE.

The 'Load What?' requester contains a list of the scene's various parts. Beside the name of each item is a box containing either the word 'YES' or 'NO'. Initially, each box contains the word 'YES'. If you click on a box with the left mouse button the contents will change from 'YES' to 'NO' or vice-versa. You can click on the boxes until they correctly show what you wish to load. For now, we want to load everything, so you only have to change the boxes if you have fiddled about with them earlier.

GETTING STARTED

Below the other boxes is one called 'Select' that affects the way that objects are loaded, normally you will wish to leave it as YES.

LOAD WHAT?	
Lamps	YES
Observer	YES
World	YES
Objects	YES
Select	YES
CANCEL	OK

The 'Load What?' Requester

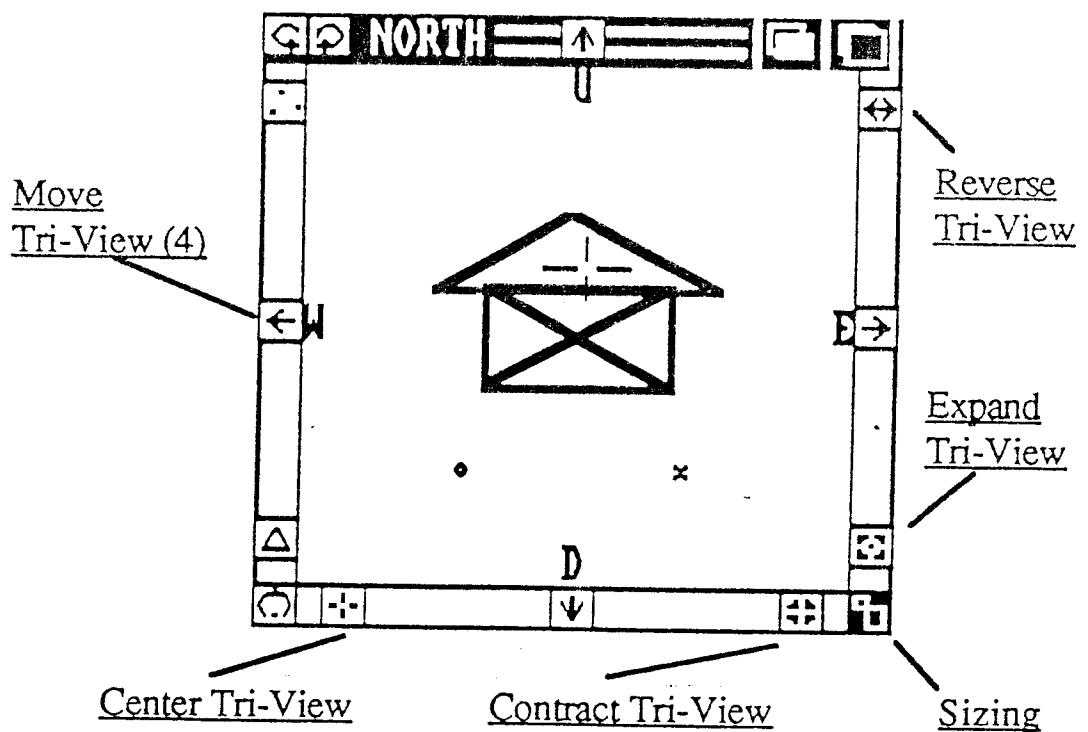
When all of the boxes contain the word 'YES', click on the OK button. After a few seconds, an image will appear in each window of the Tri-View. At first, the display may seem confusing.

This kind of picture is called a wire frame, because the object is displayed as though it were transparent, with only the edges visible. In addition to the object's outlines, Sculpt Animate divides each rectangular side with an extra diagonal line. Although this seems like a drawback, we will later see how it becomes an advantage when we are building more complicated shapes.

Look at the window with the name 'North' in its title bar. It should be at the screen's bottom left, unless you have re-arranged the windows. This is the view that faces northward. Notice that there is a small letter just within each

border of the window. At the left edge is a 'W' for 'west', to the right, 'E' for 'east.' 'U' for 'up' is at the top, and 'D' for 'down' at the bottom. These are the directions you can move things in this window.

The view in this window should be recognizable as the end view of a simple house with a ridge roof. Note that the wall seems to be braced diagonally in both directions. In fact, the north wall (the one furthest away) is braced in one direction while the south wall (the one nearest to you) is braced the other way. In a wire frame view, you cannot distinguish between near and far objects because far objects are not hidden by those nearby. Now we will look at one way to make sense of these confusing images and at the same time learn about some of the Tri-View gadgets.



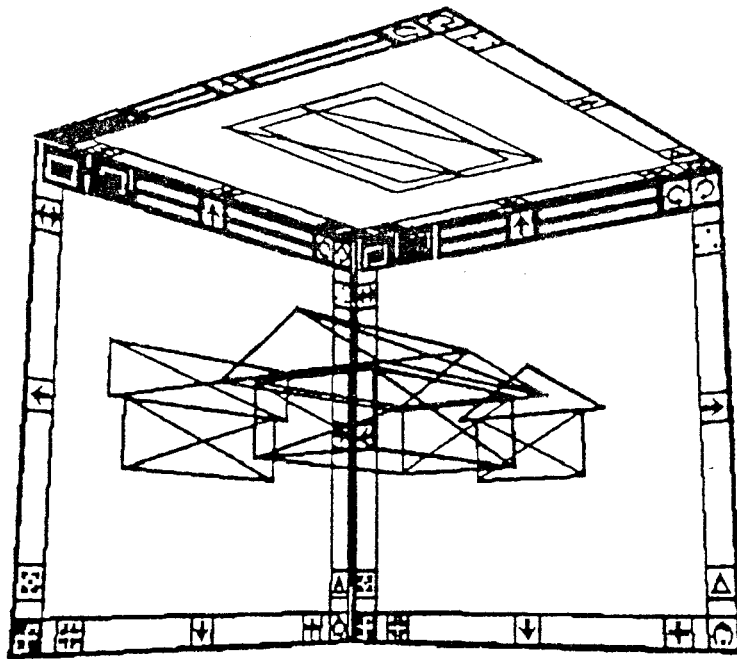
The North View

Move the pointer to the 'West' window (the window that looks west, with an 'S' for 'south' at the left side, 'N' for 'north' at the right, etc.). Unless it has been moved, it is the one at the screen's bottom right. Now look at the middle of the left hand border (south, remember?). You will see a gadget with a left pointing arrow. This is called a Move Tri-View Gadget. Click on this gadget with the left mouse

GETTING STARTED

button. After the usual screen flashing, try to see what has happened.

At first glance, the little house seems to have moved to the right (i.e. north). It's more useful, however, to say that the Tri-View has moved to the left (i.e. south). That is why the gadget's arrow points to the left. It is better to think of the objects in Sculpt Animate as fixed in space, and the Tri-View as moving. As we said before, the Tri-View windows display three views of a 'box' of space. This box itself can be thought of as a sort of window, into which we can look to see a piece of the Sculpt universe. As you might expect, it makes more sense to move the window around in the universe than it does to move the universe around in the window!



The Tri-View "Box" From Inside

If you want a 'real world' analogy, think of riding a bicycle. Most of the time it's more convenient to imagine oneself rolling down the road, rather than thinking the bicycle's wheels are moving the earth beneath you. After a while, the world of Sculpt Animate may seem just as real to you.

By now you should be able to guess what effects the arrow gadgets in the centers the other borders will have. Each will move the Tri-View in its respective direction.

Back to our example. Look again at the northward window. If you are particularly astute you will have noticed that one of the diagonal braces of the wall has vanished. This is because the brace was part of the north wall, which now lies outside the Tri-View. Click on the gadget in the middle of the right border of the west Tri-View window to move the Tri-View back to its original position. The whole of the house fits inside the Tri-View, so the second brace becomes visible again.

The Tri-View will only display objects, or parts of objects, that are within the chunk of space that the Tri-View represents. When you are dealing with complicated scenes, such as an entire village of little houses, this is a decided advantage. You can move the Tri-View to the area that you are interested in, and not be bothered by neighboring objects. By the same token, you can use the Tri-View to isolate just a part of an object, as you have already done with the little house.

Now try using the Sizing Gadget in the bottom right corner of a window to make a window bigger. See how the other windows adjust themselves in order to be consistent. Also notice that although you make the window bigger by stretching out the bottom right corner, as far as the Tri-View is concerned it gets bigger uniformly with respect to its center. If the little house was in the middle of the window before enlarging the window, it will still be in the middle after the enlargement. The image of the house does not get any bigger, there is just more space between the house and the border of the window.

There is another way of changing the size of the Tri-View. Click on the gadget that is near the bottom of the right hand border. This is called the Expand Tri-View Gadget. Somewhat to your surprise, the Tri-View windows don't expand, in fact the image of the little house gets smaller. This is relativity once again. What you have done is enlarge

GETTING STARTED

the Tri-View's 'box' so that more 'Sculpt space' is contained within it, even though the Tri-View windows are the same size on the screen. Click on the Expand Tri-View Gadget a few times to make the Tri-View even bigger (which makes the house's image on the screen still smaller). Now can you see why we like to think of the Tri-View's volume increasing, even though its windows stay the same size. If the windows had expanded this much, they would have overflowed off the screen.

The Expand Tri-View Gadget increases the volume of the Tri-View equally in all three directions, its width increases by about 40% with each click. If you hold down the left Amiga key (it is called the Commodore key on newer Amigas, but we will keep on using the old name) while the gadget is clicked, the amount of the expansion is reduced. Similarly, if the right Amiga key is held down when the gadget is pressed, each click of the gadget will have twice the usual effect. Similarly, the left and right ALT keys can be used in the same way to decrease or increase the effect of the gadget still more. Several other gadgets, such as the Move Tri-View Gadgets, can also be modified by the Amiga keys or the ALT keys.

Let's try moving around in Sculpt space a little. First, press one of the Move Tri-View Gadgets a few times until the house vanishes from view. At first, the cursor (the blue cross) will stay with the house, but when the edge of the window hits it, the cursor will be swept along in the Tri-View.

Next, expand the Tri-View with the Expand Tri-View Gadget until the house becomes visible. Point to the middle of the house and press the left mouse button. This moves the cursor to the middle of the house in that window, but look at the other windows to see if the cursor is really in the middle of the house in all views. If not, repeat the operation in another window.

Now click on the gadget that lies second from the left of the bottom border. This is called the Center Tri-View Gadget. It moves the Tri-View so that it is centered on the

GETTING STARTED

cursor's location, putting the cursor in the middle of the Tri-View windows.

Finally, click on the gadget that is second from the right in the bottom border. This is called the Contract Tri-View Gadget. It makes the Tri-View smaller, so the image of the house seems larger.

You have just completed a very useful sequence of commands, which you'll use often: Expand the Tri-View to find a point of interest, place the cursor on it, then center the Tri-View and contract it again. Except for short distances, this is often easier than using the Move Tri-View gadgets.

You can use the expand and contract gadgets like the zoom control on a camera. Expanding the Tri-View is like zooming out to a wide angle setting, the field of view gets bigger so objects seem smaller. Contracting the Tri-View is like zooming in for a telephoto effect, the field of view gets smaller and objects seem magnified. You aren't changing the size of things, just changing the way you look at them.

It is easy to get lost in the Tri-View--if you contract it too much there may be nothing visible at all. Similarly, moving the Tri-View too much will move you away from the inhabited part of the universe into a region of empty space. You can always use Expand Tri-View to 'step back' and figure out where you are.

When you run the program, the Tri-View always starts out at the center of the Sculpt universe, and it always starts out the same size relative to the universe. You can always get back to this setup if you get lost. To return to the center of the universe, hold down the right Amiga key and click on the Center Tri-View Gadget. To make the Tri-View its original size, hold down the left Amiga key and click on the Center Tri-View gadget, the Tri-View windows will be redrawn in their original size and the effects of any expansion or contraction will be removed.

Let us look at one more Tri-View Gadget before moving on to more interesting things. This gadget is called the

GETTING STARTED

Reverse Gadget, and it lies just below the top of the right border. It performs a mirror image swap of the particular window. Alternatively, you may think of it as reversing the direction of view, looking at the opposite side of our 3D box. A north view is turned into a south view, for instance. Our little house is so boringly symmetrical that you may not even notice the reflection, except that the letters indicating the direction near the left and right borders change places. With a more complicated object, this gadget can be more useful.

Making a picture

If all that Sculpt Animate could do was to manipulate objects in wireframe, it would be no more than a three dimensional computer aided design package. But Sculpt Animate has extensive image generating capabilities.

Sculpt Animate makes use of a special object called an observer. The observer can be placed at any location in a scene, and can be instructed to look in a particular direction. In addition the observer can employ one of several imaging techniques. The acts of the observer are akin to photography, and to follow this analogy the observer can be equipped with different camera, lens, and film combinations.

Let us look at how the observer operates by providing instructions to make a picture of the little house. First we must pick a location from which to view the house. Photographers know that to take a picture of an object with a regular lens, they must stand far enough back so that the object's image fits within the film size.

To make room for our observer to move back, use the Expand Tri-View Gadget until the house appears to be about one eighth of the size of the window. Going to the downward looking view, position the cursor at a reasonable distance from the house. Check in one of the other windows to see that the cursor is at some sensible distance above ground level, about as high as the top of the wall will do fine. Move the cursor, if necessary. When the cursor is where you want it, hold down the right mouse button to

bring up the menus. Go to the OBSERVER menu and pull the pointer down to the word LOCATION. Release the button.

If you look at the Tri-View closely you will see that a tiny circle has been drawn at the cursor location. That is the observer! It will be easier to see if you move the cursor away. The observer will stay fixed until you next perform an OBSERVER LOCATION command from the menus, or you load another scene from disk.

The next thing to consider is in what direction the observer's camera is pointing. Although the observer is not equipped with feet, you will be disappointed with the picture if the camera is pointing downwards. Rather than have you specify camera bearings in degrees, Sculpt Animate just lets you specify a target position. The observer then points the camera at the target so that it lies in the center of the image. A good target would be the center of the house. Move the cursor there, checking in two windows to see that the position is correct. Hold down the right mouse button and select TARGET from the OBSERVER menu. Although you may not be able to see it for all the clutter of the house, a small x will be drawn at the target location.

For the observer's camera to record an image, there has to be light. Some background illumination is provided as a default, but this gives a flat, uninteresting image by itself. The primary light source in most Sculpt Animate scenes is one or more lamps. To put a lamp in this scene, place the cursor somewhere near the observer, a little to one side and above will do, then hold down the right mouse button again and select LAMP from the EDIT ADD submenu. A small white star should appear at the cursor's position. Lamps shine in all directions, so you don't have to worry about pointing them, and the camera's exposure setting will automatically be adjusted to account for the extra light.

Now you can instruct the observer to start taking a picture. Go to the OBSERVER menu and select START. A tiny picture will appear near the bottom of the screen, and after a few seconds will start 'developing'. If you have not

GETTING STARTED

accidentally changed any of the observer's attributes, then the picture size will be TINY and the image type will be SNAPSHOT. These settings will be explained a little more later, and fully in the next chapter.

While the image is rendering ('developing'), and after it's finished, you may want to move it out of the way. You can do this as follows. Point to the picture and first press the left mouse button. Release it and press the right mouse button. The picture should vanish.

The picture is still in the computer's memory, however, and can be recalled by going to the PROJECT menu and selecting SHOW IMAGE. This brings the Amiga 'screen' on which the image was rendered 'to the front.'

Let us try a different image of the same scene. Go back to the OBSERVER menu and select MODE PAINTING. This mode renders more quickly than SNAPSHOT, so we can afford to make the image larger. Go to the OBSERVER menu again and select IMAGE SIZE FULL. Finally, select OBSERVER START.

Saving an image

The image that you have just rendered can easily be saved to disk. Hold down the right mouse button and go to the PROJECT menu, follow down to SAVE and then across to IMAGE. Then release the button. A File Requester will appear on the screen, asking for the name that is to be used when the image is saved on the disk.

If you want the data to go to a particular volume or drawer, click on the appropriate names in those listing boxes. You will probably want to use a new name for this file. Next to the heading 'File:' over the files listing is a box called a 'text entry gadget.' You may click on this box, then type in a new name for the image. When you hit the RETURN key, the image will be saved.

GETTING STARTED

If you want to save over an old image, just double-click that image's name in the file listing box. Since the file already exists, the program will ask you to confirm that this is really what you want to do. Saving data using an old name will cause the old data to be destroyed.

Images are saved in a standard 'IFF-ILBM' file, which may be loaded into some other Amiga programs, such as paint programs, etc. Sculpt Animate supports a wide range of image types and sizes, though, so not all existing programs will be able to use all Sculpt Animate images.

Images are not given icons. You may copy, list, and delete images using the Amiga's CLI interface. All image files are given the name extension '.image' by Sculpt Animate. This is how the program keeps track of what files are images, what files are scenes ('.scene'), etc.

Saving a scene

The contents of the Tri-View can be saved to disk with the command `PROJECT SAVE SCENE`. This requester works the same as the `SAVE IMAGE` requester. After choosing a file name with this requester, a second requester will appear, asking you what parts of the scene you wish to save. This is very much like the requester that appeared when you loaded the little house. Usually, you will just leave all the upper gadgets set to 'YES' and the 'which?' gadget set to 'ALL.' The meanings of these gadgets will be described in chapter 6.

Leaving Sculpt Animate

If you wish to perform other tasks while Sculpt Animate is still running, you can push its screen back with the Back Gadget at the top right of the screen. You are then free to use the Amiga's multitasking capability, as long as your machine has enough memory.

Be sure not to forget that Sculpt Animate is running, however! You will waste memory if you start it twice.

To exit Sculpt Animate completely, execute the menu command PROJECT QUIT. Because you may have some valuable data, scenes or images, that you have neglected to save to the disk, Sculpt Animate will ask for confirmation before actually stopping execution.

If You have problems

It would not be too much to hope that everyone who ever uses Sculpt Animate is able to do so without any trouble, whatsoever. It would probably be too much to expect, however. This section attempts to provide some guidance on how to deal with trouble, should it occur.

If you have been successfully using Sculpt Animate to follow the text up to this point, then you have already cleared some major hurdles--especially if you are new to computers. Your Amiga is plugged in and connected together properly, and probably free of any grossly debilitating defects. You have begun to master the boot-up process, the Workbench screen, and the vital process of backing up floppy disks. Furthermore, you have managed to run Sculpt Animate and operate some of its menus and other controls, and to grow somewhat accustomed to the style of this manual. In short, you have gained a valuable foothold on the rocky slope of the learning curve, and you can skip right now to the next chapter. This is by far the most likely circumstance.

If, on the other hand, your efforts so far have not been entirely successful, do not be too discouraged. Rest assured that many people have been using this program and its predecessors for some time, with good results. Any problem you may be having is only a setback, not a defeat. What follows is essentially a short course on troubleshooting any computer problem, as applied to Sculpt Animate.

If your Amiga is simply not working, don't fail to check the obvious--be sure all of its connectors are properly

GETTING STARTED

plugged in. Even the most experienced computerists have, at one time or another, been tripped up by a loose cable. Other problems might be a bad system disk, or perhaps even a hardware problem. Don't hesitate to call your dealer if you can't find the answer in your Amiga's manual.

If you have trouble backing up the Sculpt Animate disks, or if the program won't load, it could be a problem with either the distribution disk or with your Amiga's disk drive. The first thing to try is using the disks in another drive, either your second drive or a friend's drive. If the program works in another drive then the hardware is probably at fault. Otherwise, the program disk may be flawed, perhaps because of heat, cold, X-ray or magnetic damage incurred in shipping. If the disk seems to be at fault, read the warranty statement that was enclosed with the program, it will explain how to obtain a replacement.

Finally, if the program runs, but doesn't behave as you expect, there are a number of things that could be wrong. We will consider these one by one.

Of course, there is the possibility of bugs (errors) in Sculpt Animate's documentation or software. Few, if any, programs escape this problem entirely, but it is always the goal. Every effort has been made to ensure that Sculpt Animate is bug-free, and we continually hunt for bugs.

If you encounter a situation in which the program behaves in a confusing manner, contrary to the documentation, please write down step-by-step instructions telling how to re-create the problem and send them to Byte By Byte (Attention: SA4D Bug Reports). Be sure to include specifics about your Amiga's model, peripherals, enhancements, and system software. Also note the Sculpt Animate version number you are using; this is displayed at the top of the screen when you run the program.

We would also like to hear of any changes or additions you would like to see made to the program (Attention: SA4D Enhancements). Updates and new releases will be made available periodically.

GETTING STARTED

Before you file a bug report, read the remainder of this section. It describes the majority of problems that users have had in the past, and could provide a quick solution to your particular problem.

First, the obvious. It is very possible that the program simply doesn't do what you expected it to. Sculpt Animate is a remarkable program in many respects, and it does things that have never been done on personal computers before. To do these things, it must behave differently than, for instance, a paint program. The nature of solid object modeling on microcomputers makes it impossible to actually 'draw' in 3-D, the way you would draw on paper.

If you encounter confusing behavior, read the related part of this manual carefully, remembering that every method was designed for a specific purpose, with the Amiga's limitations in mind. Even if what a function does is not quite what you expected, in most cases it is the best or only solution to a difficult problem. Never forget the value of simply doing something else for a while--the problem will usually seem simpler the second time you look at it. If you simply cannot fathom some aspect of the program's use, consult the support policy statement that is enclosed with the program for information on how to obtain special help.

At times Sculpt Animate may seem to take an extremely long time to do something. This is difficult to pin down, because different scenes in Sculpt Animate can present entirely different problems to be solved. Extremely simple scenes, such as the example in this chapter, should never take more than a few minutes for any operation. If you increase the size and complexity of your scenes gradually, then you will surely develop an idea of how long things should take. The following chapters explain some of the factors that can make a scene develop more or less quickly.

There are also times when the amount of RAM memory in the system is inadequate for a particular task. This situation is described more fully in the next chapter ('Memory usage'). Also, users working with floppy disks

CHAPTER 3

THE OBSERVER

In the last chapter you were briefly introduced to the observer, a feature of Sculpt Animate that controls how images are produced. In this chapter we will look at the observer in more detail and describe the different kinds of images that can be produced.

Imaging modes

In order to record images, a real-life observer has to have a camera, and the camera needs film. What kind of film determines the quality and resolution of the resulting image, as well as how quickly it can be captured. Although your computer doesn't use film, there are still a number of different ways for it to 'take a picture' of the objects in your scene. These methods all differ in speed and realism, just as various types of film do. The MODES submenu in Sculpt Animate lets you set the kind of 'film' the Sculpt Animate observer will use.

The first mode in the menus is WIRE FRAME. This is the highest-speed rendering mode, but the lowest in quality. Wire Frame yields two-color images that closely resemble the objects seen in the Tri-View, with one significant difference. You see, while Tri-View windows display a flat 'orthographic' projection of their views, Wire Frame shows you a true 'perspective' view, as seen from the observer's viewpoint. Only edges are shown, but each edge is shown in the same place it will appear in the more sophisticated modes. This makes Wire Frame an excellent mode for taking a look through the observer's 'viewfinder' before rendering in a slower mode. The command EDIT MODIFY WIRE-FRAME COLORS can be used to change the colors used in wire frame renditions.

THE OBSERVER

The rest of Sculpt Animate's modes render objects as solid shapes in multiple colors. The simplest way to do this is to display each visible portion of the objects, with color and shading determined by the specified light sources. Sculpt Animate calls such an image a SKETCH or a PAINTING. These modes include the effects of perspective and the resulting pictures look reasonably solid and realistic.

As we shall see in chapter 4, the objects in the world of Sculpt Animate are made up from triangular pieces. In these two modes, the image of each piece is 'painted' with a single color. While this can yield good results, the image is not strictly accurate. For instance, if one end of a triangle lies closer to the light than the other, it will not seem more brightly lit, as it should. Most objects will also have a faceted appearance because adjacent faces cannot be 'blended' together.

SKETCH mode differs from PAINTING in two ways. Firstly it is faster, but a price has to be paid for the speed, so SKETCH is also less accurate. Sometimes one object that should be hidden by another will be incorrectly drawn in front. This is because SKETCH mode uses a naive way of determining which of two objects is in front of the other. An average distance from the object to the observer is calculated; objects with closer average distances are assumed to lie in front of other objects. This is not always correct, so some images produced in this mode will be flawed. The speed advantage makes SKETCH an attractive way of previewing a scene in any case. PAINTING mode resolves depth ambiguities by breaking up some faces into smaller ones. This does well, even for faces that intersect, but it requires extra calculation so PAINTING takes longer than SKETCH to generate an image.

The SCANLINE PAINTING mode yields results that look similar to PAINTING, but it employs a different approach to creating the image. Briefly, it sorts faces much as PAINTING mode does, resolving intersections. SCANLINE PAINTING does this pixel by pixel, however, which can improve speed as scene complexity grows. SCANLINE rendering's advantage increases with objects

that cover a relatively small portion of the image, or objects that have many intersecting faces. Its output is similar to that of PAINTING (hence its name).

These three rendering modes are fairly limited in the number of colors which they can display. This is because they use the Amiga's 'normal' video display capability of 16 or 32 colors in any image (or fewer). Images rendered in this mode can be written to disk and subsequently used in almost any Amiga 'paint' program.

Sculpt Animate's higher-quality imaging modes all use the Amiga's Hold And Modify (HAM) facility. The mechanics of HAM are rather complex, but its effect is to let these modes display as many as 4096 different colors on the screen at once, at the cost of some horizontal resolution. With all these colors available, it is possible to display images with subtle shadings and blends of color.

A HAM image's many colors are derived from a 'map' of only 16. This can make color transitions difficult, so sometimes a false color fringe will be seen between areas of different color. As soon as Sculpt Animate finishes rendering an image, it will pause for a time as it finds the best possible map of 16 colors to minimize these fringes, then the image will be re-drawn with the new color map.

The fastest imaging mode to use HAM is SCANLINE SNAPSHOT. This mode uses the additional colors to enhance realism by shading color smoothly across faces, and, if desired, smoothing adjacent faces together. A face may also be made 'shiny', so that specular highlights will appear on its surface wherever appropriate, as determined by the scene's lighting. Color, smoothing, and surface texture may be set differently for various faces. This will be discussed in Chapter 4. Only the DULL and SHINY textures are used in SCANLINE SNAPSHOT.

The next step in increasing realism is to use a technique called ray tracing, which traces rays of light from the observer's eye to their source, computing an intensity and color for each pixel on the screen. A pixel ('picture

THE OBSERVER

element') is the smallest area of the screen that can be given a distinct color and brightness. The number of pixels in an image will vary according to its size and resolution, as will be discussed later in this chapter, but a typical image might have hundreds of thousands of pixels; meaning hundreds of thousands of rays to be traced to an indefinite length. Understandably, ray tracing taxes the Amiga's computing power.

The rewards are worthwhile, though. Like SCANLINE SNAPSHOT, ray tracing can determine shading and color smoothly across flat surfaces, and smooth rough edges. Ray tracing can go still farther, though. Since each pixel's light value is traced all the way back to its source, rays can be traced through transparent objects, and off of reflective surfaces, automatically creating accurate and realistic images of objects that would be very difficult to draw or paint. Uncanny realism can be achieved.

Sculpt Animate calls its simplest and fastest kind of ray traced imaging a SNAPSHOT. Lacking only shadows, this mode can still create very realistic images, rivalling photography in some cases, were it not for display limitations. Still more realistic is PHOTO mode, which shows all the ray tracing prowess of SNAPSHOT, plus distinct, realistic shadows cast by objects on the ground and each other. PHOTO mode's extra calculation incurs an extra time penalty, though, so if shadows aren't likely to be a big part of your image, feel free to use SNAPSHOT or SCANLINE SNAPSHOT.

Whatever mode you use, the time taken to generate an image depends upon the complexity of the scene and the size of the image, as well as the imaging mode. The time needed can vary from a second or two for a WIRE FRAME of something simple to many hours for a difficult PHOTO.

With SCANLINE, SNAPSHOT and PHOTO modes, it is a good idea to try things out with a TINY or SMALL image size, because the rendering time varies with the size of the final image. Of the three HAM imaging modes,

SCANLINE SNAPSHOT is generally fastest, although SNAPSHOT may be faster for certain simple scenes.

Now that you know about the various imaging modes, you need to know how to select one. You can specify which is to be used with the OBSERVER MODE menu command (from now on in this book, we will assume that you have mastered the operation of the menu). The MODE submenu lists the choices with a checkmark to show which mode is currently active. If the marked item is not the one you want, select the correct menu item in the usual way. The checkmark will move to reflect the change.

The OBSERVER MODE command also lets you choose the screen resolution in both the vertical and horizontal directions. Resolution determines the amount of detail which can be displayed in a given area of the screen. On the Amiga, resolution can vary in either direction. LO-RES and HI-RES refer to low resolution and high resolution horizontally; NO-INTERLACE and INTERLACE refer to low resolution and high resolution vertically. Thus, a LO-RES, NO-INTERLACE image would be the Amiga's grainiest image, HI-RES INTERLACE its sharpest.

As is usually the case with computers, every improvement carries a cost. The highest image resolution is not available in every imaging mode. Specifically, the Amiga cannot display a 32-color 'normal' image or any HAM image in HI-RES. If HI-RES is selected, all SKETCH, PAINTING, and SCANLINE PAINTING images will be automatically limited to 16 colors. If any of the HAM modes is used, a HI-RES setting will be ignored, and a LO-RES image generated instead.

None of these limitations apply to INTERLACE, but it, too, carries a cost of sorts. Some high-contrast images will flicker noticeably in INTERLACE resolution--this is normal on the Amiga. Special 'high-persistence' monitors are available which can reduce or eliminate this problem.

Each of the higher-resolution settings doubles the number of pixels to be rendered. In SKETCH or

THE OBSERVER

PAINTING mode this will have little effect on speed, since the number of calculations depends mostly on the number of faces. In the SCANLINE and ray trace modes, however, each pixel is calculated independently, so doubling the resolution will more or less double the rendering time. In any case, the higher resolution images will require somewhat more RAM and disk space.

Sculpt Animate will generally assume that you want to use as many colors as possible in a given imaging mode. The command OBSERVER MODE BIT PLANES gives you some control over the number of colors that will be used, but you need to know how this works. Computer screens, and hence computers, create colors by mixing red, green and blue light. The brightness of each color is specified by a number. On the Amiga display, this number can range from zero to fifteen. Thus you have sixteen levels of light each for red, green, and blue, making 4096 total combinations.

In the normal (non-HAM) display modes, it would require too much memory for the Amiga to directly display all 4096 colors in an image, so a particular image may only use a subset of the available colors, 16 of them in HI-RES and 32 in LO-RES. Sculpt Animate attempts to pick the best set of colors for a given scene.

Some Amiga programs impose further restrictions on the number of colors that can be used. If you plan to use your images in such a program, you must issue the command OBSERVER MODE BIT PLANES. A requester will pop up asking you how many bit planes you desire. One bit plane permits the use of two colors. Each additional bit plane doubles the number of colors as follows:

Bit planes	Colors	
1	2	
2	4	
3	8	
4	16	
5	32	(LO-RES only)

If you specify zero or six bit planes, Sculpt Animate will use the greatest number possible for the selected imaging mode.

Bit plane values greater than six would normally have no meaning on the Amiga, but in Sculpt Animate there is one special case. By specifying 24 bit planes (equivalent to 16,777,216 colors), you can have raw red, green, and blue (RGB) image data written out to disk files.

Most users of Sculpt Animate will never have any use for raw RGB data, so you will probably want to skip to the next section. The ability to write RGB images is provided so that specialized hardware and image processing software may use Sculpt Animate images. For the time being, then, such things are primarily of interest to programmers and other technically inclined users. For those users, here are some specifics about the raw RGB facility.

When you ask for 24 bit planes, you will be asked for three file names, the width and height of the desired image in pixels, and each pixel's ratio of width to height. These settings will only take effect if the image is rendered with the START menu command, and if the imaging mode is SCANLINE (either kind) SNAPSHOT, or PHOTO. RGB data for animations must be handled through a frame buffer device driver, as explained in chapter 12.

Three files will be written, each will contain an image of the scene in either red, green, or blue. The data in each file is stored as one eight-bit brightness value for each pixel. If fewer bit planes are required, your special-purpose software will need to mask off the unused low-order bits and pack the data as needed. The pixel bytes are stored serially, one scan at a time, beginning with the pixel at the upper left hand corner of the image. There will be no header information, just data. Be sure there is enough disk space for each file, a 704 by 480 image will need 337920 bytes for each file. You may type different path names into the three filename requesters.

THE OBSERVER

The requested width and height values refer to the width and height of the image, in pixels. This depends on your application, which is up to you. Except for storage, there are no known limits to the height values which may be entered. The maximum width should be 32768 pixels--values up to 1024 have been tested with no problems.

The pixel width and height values refer to the aspect ratio of pixels on the screen which is to display the image. For instance, a 100 pixel square on an Amiga screen is usually about one tenth higher than it is wide, so its width:height ratio would be 10:11.

Placing the observer

In the last chapter, we explained how to change the observer's position, but we will repeat it again (in case you look here first). First you must select a position in the Tri-View with the cursor. You may need to make the Tri-View bigger, so you can place the observer far enough back to get a reasonable view.

When the cursor is at the right place (look in at least two windows), go into the menu and select OBSERVER LOCATION. When you execute this command, a small circle will appear in each window, to mark the observer's location.

Aiming the observer

The observer must be told which way to look. Although you may know what is the most interesting object in the world of Sculpt Animate, the observer doesn't. As we explained in chapter 2, the observer always points towards a special spot called the target. This is specified in a similar manner to the way that we picked the observer's location. We move the cursor to the target position and select OBSERVER TARGET from the menu. A small x will mark the spot. The target is usually placed near the most

interesting part of the scene, so that part will appear in the center of the image.

Never be timid about using the observer as an editing tool. As you continue with this manual, you will learn how to create objects, color faces, and arrange scenes. It will often be helpful to move the observer and target in order to view a point of interest, then 'take a look' with either the WIRE FRAME or SKETCH imaging mode.

Camera lenses

We have already drawn an analogy between the observer and a photographer wielding a camera. If the camera is equipped with a zoom lens, it can zoom in for a telephoto shot where a small angle of view fills the whole picture. Alternatively the camera can zoom back so that the field of view is wide. In a similar way, the observer can be supplied with a lens of your choosing. Use the OBSERVER LENS menu command to examine and change the current selection.

The observer has three standard lenses that should suffice for most scenes. Alternatively, you may select the OBSERVER LENS SPECIAL option. When you select this, a requester will prompt you to enter the focal length of a lens of your own design. The smaller the number that you enter, the wider the angle of view, and the smaller an object will appear. For instance, of the three standard lenses provided, the NORMAL setting corresponds to a SPECIAL setting of 50, WIDE-ANGLE corresponds to 28, and TELEPHOTO corresponds to 135.

In general, these numbers are analogous to millimeter focal lengths of lenses on a typical 35-mm camera. If you enter a number in the telephoto range of 135 or larger, distant objects will appear larger than normal, with a reduced sense of perspective. If you enter a very small number, such as 10, then you will get a fish-eye effect with grossly exaggerated perspective.

THE OBSERVER

Astute photography buffs will notice that the camera analogy fails slightly, especially when a fish-eye lens is used. This is a rather obscure artifact of geometry, so you can ignore the explanation and skip to the next section if you like.

In a 'real' photograph, long parallel lines will tend to curve towards each other near the edge of the image. This is because camera lenses have a finite 'aperture' through which light is admitted, and the light is refracted unevenly across this aperture. In Sculpt Animate, the observer's lens has an infinitely small aperture, it is 'perfect', so no such distortion occurs. This difference also gives the observer's lens an infinite depth-of-field, so objects can never be out of focus.

So which is 'right', the real lens, or Sculpt Animate's? Both, and neither. Both lenses have their advantages. The real lens's field distortion and limited depth of focus can be used to create exciting effects, but the total removal of these characteristics presents the Sculpt Animate user with a creative situation that never existed with a 'real' lens, and some exciting new results are bound to arise.

To say that either kind of lens is more 'real' is really incorrect. When their images are intended to resemble what the human eye would see, they both do pretty well, but neither is perfect. The only time when their results differ greatly is when they are used to create effects that look 'unreal' anyway.

Exposure control

Most modern cameras have an automatic exposure control so that pictures don't come out too dark or too light. Sculpt Animate simulates this feature. This means you usually don't have to worry about exposure when you add or move lamps in a scene.

Like the automatic exposure control on a camera, most of the time Sculpt Animate does a good job of controlling the brightness of the image. It works according to a simple rule:

The object in a scene that is closest to a lamp will be brightly lit, but not lit so brightly that it washes out.

Sometimes, perhaps for artistic reasons, you may want to change the exposure. Select `OBSERVER EXPOSURE MANUAL` and a requester will pop up asking you to override the automatic setting. Enter a number as a percentage of the standard value. If you want your picture to be twice as bright as normal, enter 200. Because the Amiga screen has a very limited range of brightness, the image that you get may look flat and washed out. Similarly you can enter a number less than 100 if you desire a gloomy and somber effect.

The Jaggies

Computer images are built up from discrete pixels. Even with 640 by 400 pixels, the image is still ten times more coarse than a 35mm slide. The regular layout of the pixels makes them evident to the eye as stair-step bumps, particularly along shallowly slanted borders between two very different colors. Engineers and other technically-minded folk call these obtrusive artifacts 'aliasing', but most people just call them 'the Jaggies'.

When images are rendered in HAM mode, it is possible to use a technique called anti-aliasing to reduce the effects of the jaggies. Anti-aliasing hides the jaggies by blurring sharp color transitions slightly. It will increase rendering time somewhat. The command `OBSERVER ANTI-ALIASING` can be used to control the type of anti-aliasing that is employed.

Which is better--blurring or jaggies--will usually depend on the subject matter and your own taste. You should probably try an image with and without anti-aliasing to see which you prefer. Images with lots of high-contrast color transitions will often need anti-aliasing, especially when they are used in animations.

Dithering

Although the Amiga display hardware can utilize 4096 colors, this is not enough if you want the highest possible image quality. Each of the primary colors can only take on one of sixteen values, so colors jump by almost 8% from one level to the next. The human eye can easily detect this transition as 'color bands' in the finished image. In the HAM imaging modes, Sculpt Animate usually employs a blending technique called 'dithering' to hide these transitions. Dithering works well, but it adds graininess to the image.

You can control this graininess with the command `OBSERVER ANTI-ALIASING DITHERING`. A requester will pop up asking what percentage of standard dithering is needed. A value less than 100 will reduce the graininess at the expense of the color jumps being more obvious. A value of zero will eliminate dithering entirely, causing distinct color banding. Values greater than 100 cause more dithering to be applied.

Image size on the screen

You can select the size of image that Sculpt Animate creates. Usually you will want the image to occupy the full size of the screen, but there are a few reasons why you may want to make a smaller image.

If you are planning to import your picture into one of the Amiga paint programs, then a small picture may be all you need, particularly if you are using the image as a brush.

Another advantage of a small image is that you can continue to work with the Tri-View, with the picture displayed at the bottom of the screen. If you point to the screen near the top of the black area to either side of the picture, and hold down the left mouse button, you can drag the image up the screen or push it down so that it almost vanishes. To do this, you have to hit the black area just right, between about four and twelve scan lines from the top.

Small images also spare your time and system resources. They will always require less memory for storage and display than larger images, and in the scanline and ray-traced modes, smaller images render much more quickly. It's usually a good idea to render a small image to check shadows and lighting before starting a full size picture.

Take note that the initial scene data processing and optimization will be the same for all images of a certain scene. Still, in the rendering phase a TINY image should be about 64 times as fast as a FULL one.

You change the picture size by selecting a choice from the OBSERVER IMAGE SIZE menu selections. The actual size in pixels of the resulting images will vary from country to country. This is because two different video standards are common: NTSC, which is used in the USA and Japan, and PAL, which is used in Europe and elsewhere. Sculpt Animate automatically adjusts both its Tri-View and images to your Amiga's video display.

Here are the image sizes (in pixels) that may be used:

Size	NTSC width x height	PAL width x height
TINY	48 x 25	48 x 32
SMALL	80 x 50	80 x 64
MEDIUM	160 x 100	160 x 128
FULL	320 x 200	320 x 256
JUMBO	352 x 220	352 x 282
VIDEO	352 x 240	352 x 290

The above values are for lo-res, non-interlaced images. If INTERLACE is employed, there will be twice as many vertical pixels, but the image size will be the same. If HI-RES is used there will be twice as many pixels horizontally, except in the HAM modes, where HI-RES has no effect.

You can force the program to use one video standard or another with the Workbench menu 'Info' facility. Do this by selecting the SA icon and calling its 'info' window. Click ADD on the 'Tool Types' string gadget on that window,

THE OBSERVER

type in either 'TVTYPE=NTSC', or 'TVTYPE=PAL', hit RETURN, and click SAVE. The setting takes effect when you next run the program by double-clicking its icon.

Tilting the camera

Many photographers have difficulty holding a camera level. You have doubtless seen photos of perfectly upright buildings that resemble the Leaning Tower of Pisa, or of people who seem to have lost one shoe. In Sculpt Animate, the observer always holds the camera straight unless told to do otherwise. To change this, you can use the OBSERVER TILT menu command to specify an angle of tilt, in degrees. A positive value will tilt the camera to the left, negative values tilt it to the right. You can use a value of 90 to create a vertical format image, or a value of 180 to make the image come out upside down.

Most of the time it is easy enough for the observer to hold the camera upright, since it's usually pretty obvious which way is up. But what if the observer is directly over the target? No one camera orientation makes more sense than any other in this situation, because 'up' isn't in the picture! In this case, the program will pick an arbitrary orientation.

What is more likely to happen in this situation is that the camera will be almost, but not quite, directly over the target. If it is a little bit east of the target, then 'up' will be west. If it is a little bit northwest, 'up' will be southeast. In other words, 'up' is always the direction opposite the observer's position. In an animation, this can make the image seem to wheel about drunkenly. A similar behavior occurs when the observer is beneath the target.

If you must place the observer above the target, you will need to do so carefully, but it's fairly simple. As we've seen, if the observer is to look downwards, the orientation of the resulting image can be controlled by placing the observer just a little bit to one side. In chapter 13, you will see how the coordinates window can be used to do this with

precision, easily. Only a tiny offset will be needed, so the image will be indistinguishable from a true downward view.

If you're surprised that 'up' and 'down' can cause us so much trouble, don't be. We're in good company, some of the largest telescopes in the world have trouble looking straight up, for similar reasons!

Starting the image making process

After you have set up a scene and the observer's attributes, you can start the picture making process with the OBSERVER START menu command. If a hidden image is already being rendered (see the next section) you will be informed, and nothing more will happen. Otherwise the image creation will commence.

Image rendering will proceed differently in different imaging modes. As you use the program, you will learn what to expect. In the HAM modes, the image will usually not be displayed until it is complete, but a status window will display the phase and progress of the calculation. You can request that, if possible, the image be displayed as it is rendered. The command OBSERVER DISPLAY EARLY does this. OBSERVER DISPLAY LATE re-instates the normal procedure.

Sculpt Animate renders a HAM image initially in 4096 true colors, then analyzes the image to pick the best possible HAM color map. If memory is low and there is enough disk space, the program will save temporary image data to disk, either in the current directory or in any mounted device named 'SATEMP:'. A message will warn you if there is not enough RAM or disk space to render the image.

Hiding and restoring the image

Once a picture has been created, or sometimes even as it is being produced, you can hide the image from view. This

THE OBSERVER

will only be possible if the normal system pointer is visible, not if the grinding gears are going.

To hide the image, first place the pointer over the image itself and click the left mouse button. This will make the mouse pointer disappear, so don't move the mouse any more or it might go off the picture. Now if you click the right mouse button the image will disappear. You will usually just want to click left-right in quick succession.

When the image disappears, the Tri-View will become visible. Before you can do anything else, you must move the pointer inside one of the Tri-View windows and click the left mouse button. This 'selects' the Tri-View, which makes the menus and keyboard active.

You can bring the hidden image back into view with the menu command PROJECT SHOW IMAGE. If you want to remove the image entirely, freeing up the memory it uses, use the menu command PROJECT UNLOAD IMAGE.

Experienced Amiga users will find that the hidden image is on its own Amiga graphics 'screen'. There can be several such screens present in the system at any one time, and there are a number of ways to bring a particular screen into view (see your Amiga documentation for details). Sculpt Animate's JUMBO and VIDEO sized images are larger than the Amiga's standard screen, however, so they will not be properly displayed unless you use PROJECT SHOW IMAGE to bring them into view.

Aborting the image making process

When the observer is instructed to start making a picture, there is a period when the full capacity of the Amiga is required. During this time, the pointer will turn into the grinding gearwheel image. As soon as the regular pointer returns, you are free to continue working with the Tri-View, even though image generation is still in progress. Changes that you make to the scene or its objects will not effect the image being generated.

If, after your normal pointer returns, you decide that you do not want the image, you can use the OBSERVER ABORT menu command to stop the processing. After an image generation has been aborted, the PROJECT SHOW IMAGE command may be used to recall the partially completed image, and PROJECT SAVE IMAGE can be used to save it. Use PROJECT UNLOAD IMAGE to free up any memory that is involved in maintaining the fragment of image.

Memory usage

If you are curious about what resources are being used, you can execute the command OBSERVER STATUS. This turns on (or off) a display at the top of the screen. Most of the items in this display are explained in the next section. On the left side of this display, the amounts of unused chip and fast memory are shown. Chip memory is required to display screens, while both chip and fast memory can be used for storing code or object data. This distinction will be explained more in a moment.

When memory gets low, the program may not be able to proceed. Sometimes Sculpt Animate will recognize the condition and simply refuse to perform a command, but at other times the memory shortage occurs at a critical point and there is no alternative but to halt the execution of the program without warning.

It is a good idea to keep track of memory usage, and save any important scenes or images at frequent intervals. Sculpt Animate is quite a large program, but all the code need not be in memory at once. When memory starts to run low, unused code is jettisoned to allow space for more important items. When memory becomes available, the code will be reloaded in anticipation of being used. You can control this automatic operation with the PROJECT LOAD CODE and PROJECT UNLOAD CODE commands. The only drawback of having code unloaded is that a delay will occur before some commands can be executed, as the code must be recovered from disk. If you are working with

THE OBSERVER

floppy disks, you will be asked to insert the program disk when this happens.

This is not the only time when Sculpt Animate will load code. The program makes use of a facility of the Amiga called 'disk loaded libraries'. Most of the time you won't need to worry about this, but if the you are ever asked to insert the Workbench disk (or whatever disk you used to 'boot up' the system), this is probably why. Of course, other programs running concurrently, including the workbench screen, may also ask for a particular disk. This is normal on the Amiga.

Just as the code can be loaded and unloaded, so can the workbench screen. The commands PROJECT UNLOAD WORKBENCH and PROJECT LOAD WORKBENCH do this. Note that the workbench can only be unloaded if it is not being used, and it can never be unloaded if a CLI window is present. PROJECT UNLOAD WORKBENCH will do nothing in these cases.

The terms 'chip memory' and 'fast memory', used above, refer to Random Access Memory (RAM), this is volatile (non-permanent) memory within the computer which can be accessed directly by the microprocessor.

Fast memory is the Amiga's general purpose memory. All expansion memory on current Amiga models is classified as fast because, theoretically, the microprocessor has free access to the memory at all times. Fast memory is where programs and object data are preferentially stored.

Chip memory may also hold these things, if there is no more fast memory available. More importantly, though, chip memory is the only place where graphics display data can be used, since it is the only memory which the Amiga's graphics chips may 'borrow' from the microprocessor. As you can imagine, this makes it very precious on a graphics-intensive machine like the Amiga.

Current models of the Amiga can have no more than 512K of chip RAM. This means that, no matter how much

expansion RAM you have, there will still be times when an operation cannot be performed, especially when you generate images with HI-RES, INTERLACE, and JUMBO or VIDEO'set. You may need to make it a habit to do an UNLOAD WORKBENCH and make the Tri-View windows as small as possible when working with such large images.

Users in Europe, Australia, and other countries that use PAL video may not be able to use PROJECT SHOW ANIMATION (chapter 12) on animations rendered in these large resolutions. These animations will require the stand-alone playback program Movie, included on one of the Sculpt Animate distribution disks. See the Movie documentation file for more information.

It is reported that conversion kits will eventually be available to upgrade the Amiga 500 and 2000 to one Megabyte or more of chip RAM.

Other status display items

As the last section explained, the command OBSERVER STATUS calls up a program status display. In addition to the free memory readout already discussed, the status display also lists how many vertices (selected and unselected), edges, faces and lamps are in use. If an image is being rendered, a rough estimate of the remaining time is also displayed.

The vertex readout can be extremely useful. It is easy to count vertices just by selecting an object, or part of an object, and looking at the status display. This is especially useful in creating motion paths for use in animations (see chapter 10). The edge and face readouts are primarily useful as an indicator of how complex the scene is. The more faces there are, the longer the scene will take to render. It is good to be careful not to use any more faces than necessary.

NOTES

CHAPTER 4

BUILDING OBJECTS

In this chapter we will describe the way that Sculpt Animate represents solid objects, and how you can design any shape with your mouse.

Objects in Sculpt Animate are built up from individual triangles. In the Tri-View, each triangle is visible as three edges joined at three vertices. In this chapter we will describe the lowest level of object building, dealing with the triangles on an individual basis.

It would be extremely tedious to build a complicated object by constructing all of its triangles one-by-one. In chapter 5 we will be using some power tools that enable you to conjure up complicated shapes with hundreds of triangles, using only a few strokes and clicks of the mouse.

You need to know how the low-level construction is performed, though, to be in a better position to use these power tools. The low-level operations can also be used to expand upon objects that are hewn by more powerful techniques, increasing the total number of objects that may be created with ease.

Vertices

The simplest entity that Sculpt Animate can handle is called a vertex (the plural of vertex is vertices). Vertex is just a fancy name for a point at the corner of a triangle. We had better not call it a point, though, because that word is already overused, we might otherwise have to say 'point the pointer at the point!'

If you remember how we located the observer within the Tri-View, you will not be surprised at how we create a vertex. First we position the cursor at the exact spot where

BUILDING OBJECTS

we would like a vertex (by pointing with the mouse and clicking the left mouse button). Then we issue the command EDIT ADD VERTEX.

There is an easier way to do this, just move the cursor to the correct place and, while still holding down the left mouse button, click the right mouse button once. Instead of bringing up the menu, clicking the right button with the left button down will create a new vertex. It is portrayed in the Tri-View by a bright yellow dot.

Try making some vertices, but at this stage do not make more than three of them, we will see why in a moment. Create the three vertices in a triangular pattern about half the size of the Tri-View.

Edges

The next step in low level construction is to build edges. Edges always end at a vertex. Indeed, the main purpose for having vertices is to define the position of edges.

If you succeeded in creating three vertices, there is a Tri-View gadget that you can use to join the vertices. It is located near the bottom of the left hand border of each window. Its icon is a triangle. Click on it and see that dark lines are drawn between the vertices. This is how the Tri-View portrays edges.

If, for some reason, edges were not created, and the screen simply flashed at you when you clicked on the Edge Building Gadget, it was probably because you had created more than three vertices. This can happen if you accidentally double-click the right button, creating two vertices in a spot instead of one, or if you have already been playing with the mouse. You can clear away the vertices by using the EDIT ERASE ALL menu command. Create three new vertices and try again to construct the edges.

Why is the Edge Building Gadget so persnickety? If you had only created two vertices, it would have happily joined

them with a single edge. If you had a dozen vertices, the Edge Builder would have had to make 66 edges to join all combinations of vertices. This might make a credible image of a bird's nest, but it would have little other use. This is why the Edge Builder will only work with either two or three vertices, building one or three edges, respectively.

The command EDIT ADD EDGES has the same effect as clicking the Edge Building Gadget.

Faces

The next, and fortunately last, element that we must consider is called a face. In Sculpt Animate all faces are triangular. There is no need to provide a gadget to create faces, for Sculpt Animate has a rule: whenever three vertices are joined by three edges, a face is automatically created. So, unbeknownst to yourself, you have already built a face. It is hardly surprising that you did not notice, because faces are not depicted in the Tri-View. They don't have to be shown, because whenever edges make a triangle, a face will be there.

Although faces are not rendered in the Tri-View, they are the only parts of an object that are seen by the observer when making an image in any mode except WIRE FRAME. Vertices and edges are never seen in the final picture.

To prove this, go to another window. Your triangle will probably look like a straight line in this window, because it is being viewed edge on. Move the cursor away at right angles from the line--expanding the Tri-View, if necessary--and locate the observer at this point using the OBSERVER LOCATION menu command. Next, move the cursor back to the triangle, checking in two windows to be sure of its position in three dimensions. Then use the OBSERVER TARGET menu command to point the observer at the triangle. Start the image making process with OBSERVER START.

BUILDING OBJECTS

The picture that will be created will hardly be a work of art. Indeed, even if you have done everything exactly right, you will only see the dark silhouette of a triangle. This is because we forgot to turn on a lamp. Move the cursor back to the vicinity of the observer and a small distance to one side. To create a lamp, execute the EDIT ADD LAMP menu command. This time, if you make a picture with the OBSERVER START command, the triangle should be flooded with light. If the lamp was sufficiently to one side and you are in SNAPSHOT or PHOTO mode, you may even see a hint of shading across the otherwise bland triangle.

This really does seem like a lot of bother just to draw a triangle. Fortunately, we will soon be using tools that will mass produce faces so that you can sculpt shapes of any kind.

Selecting vertices

Sculpt Animate uses a variety of methods for choosing vertices with which to work. It distinguishes between selected and unselected vertices. As a general rule, a vertex must be selected before you can manipulate it in any way (the exceptions: 'connected' commands such as SNAP CONNECTED in chapter 4; and spline manipulation in chapter 8). A selected vertex is portrayed as a bright yellow dot. The vertices that you created by holding down the left mouse button and clicking the right button were shown in yellow, so they were selected vertices.

To change the state of selection of a vertex, you must point at the vertex with the pointer and double click the left mouse button. Try it with one of the vertices you've made in the Tri-View. The color of the vertex will change from yellow to a dull purple. If you double click again, it will change back to yellow. Selected vertices are yellow and unselected vertices are purple.

In order to change a vertex, the pointer must be within a few pixels of the correct spot. If Sculpt Animate cannot find

a vertex in the immediate vicinity, no vertex will be changed, but the screen will flash briefly to warn you that you missed. If this happens, move the pointer closer to the vertex and try again.

We say that double click selection works on the 'indicated vertex'. Several of the commands you will learn operate on the indicated vertex. If you want to indicate something, you do not usually have to position the cursor in all three dimensions, two dimensions suffice to make your wish known. In other words, you only need to indicate a thing in one window.

The exception to this is when two or more vertices are close to the cursor in the current window. If this happens, the vertex among them which is closest to the cursor in the other windows will be chosen. Double click selection will cause the cursor to jump, in the other windows, to the location of the desired vertex. Make it a habit to look at another window anyway, to see that what you wanted is what you got.

Now that you know about selection and deselection, we can tell you a little more about how the Edge Building Gadget works. It will construct edges between the selected vertices, provided that either two or three vertices are selected. If some other number of vertices is selected, the screen will flash a warning and nothing else will happen.

You now have enough information to create a triangular net of vertices and edges. Turn off the selection of all but two vertices and use the Edge Builder to create a new edge. Remember that each time you complete a triangle with three edges, a face is constructed, and it is these faces that are visible when you use the observer to generate an image.

If you move from window to window, your vertices will not all lie in one plane and you can construct solid shapes. Try to construct a pyramid, and then see what it looks like to the observer. Don't worry if it isn't perfect, the exercise is what's important right now.

BUILDING OBJECTS

Remember that you have to have at least one lamp suitably placed or your constructions will just look like a dark hulk. To create a lamp, position the cursor at a suitable spot, preferably behind, above and to the side of the observer, and execute the command EDIT ADD LAMP. If you don't like the lamp where you've put it, for now just erase it with EDIT ERASE ALL LAMPS, and add a new one. You'll learn how to move lamps at the end of this chapter.

If you succeeded in fabricating a pyramid, try to build a cube. Here's a tip that can help keep it square: if you want two vertices to 'line up' squarely in the Tri-View, place your first vertex, then move the cursor to about the right place for the second vertex. In one of the other windows, the cursor will seem almost centered on the first vertex. In fact, if you go to the other window and center the cursor directly on the first vertex, you can make your second vertex and it will be right in line with the first! In this way, you can use the Tri-View as a carpenter's square. The precision is limited by the screen, of course--chapter 13 explains some much more powerful tools for precise placement of vertices, etc.

Back to the example at hand. Once you've laid out the cube's basic shape and connected the corners with edges, you still haven't made it a visible object. In order to give it visible faces, you will have to divide each of the square sides with an extra diagonal edge, making two triangles. If you wanted to make something that looked like an open box, you could leave out the diagonal on the top side. Try doing this, and look at the resulting image.

More than likely, what you will see is an open box, but there is a chance that you'll see something else, even if you've done everything correctly. Enough suspense--it just so happens that if a box is constructed in just the right (or wrong) manner, it will have one or more 'hidden faces' inside it. You see, it is possible to create your diagonal edges (the ones that make the sides visible), so that three diagonals join at three vertices to make an extra face inside the box. This usually won't make much difference, but remember that it can happen--just in case. The next chapter

will tell you how to get Sculpt Animate to build a basic cube for you. It'll usually be best just to start with one of those and alter it to suit your purpose.

Now that you have a couple of more examples under your belt, it's time to learn some more features.

Another way to select vertices is with the Select Gadget. It can be found near the top of the left border of a Tri-View window, and its icon is three dots. This gadget changes the selection state of vertices that are inside the Tri-View--*only* the ones visible in the Tri-View, no others. We will make use of this property in a moment.

Here's how the gadget works: if one or more vertices in the Tri-View are selected, then the gadget deselects the vertices. If no vertices in the Tri-View are selected, then the gadget causes all of them to be selected. You can use the gadget to toggle all the visible vertices between selected and deselected.

A useful trick, when you wish to select only some vertices of an object, is to move, expand, or contract the Tri-View until only those vertices that you wish to change remain inside. The Select Gadget will then change only these vertices. This is an excellent way to select one side of a cube, the end of a wing, the front or back of some letters, etc.

There are still more ways to change the selection state. The menu command EDIT SELECT ALL and EDIT DESELECT ALL can be used to select or deselect every vertex in the scene, not just those in the Tri-View. The commands EDIT SELECT CONNECTED and EDIT DESELECT CONNECTED operate on all vertices that are connected to the indicated vertex. Any other vertex that is linked by an edge, either directly or indirectly, to the indicated vertex is considered to be connected to it.

As an example, if you were to reload the little house and try out the SELECT and DESELECT CONNECTED commands, you would find that the roof was connected

BUILDING OBJECTS

together, but was not connected to the walls. All four walls are connected together. This makes it easy to select the roof, without changing the walls. Later in this chapter we will see how to literally raise the roof.

Another useful command for selecting vertices is EDIT SELECT SWAP. As its name suggests, all selected vertices become unselected and all unselected vertices become selected. This command works on all vertices, not just those that lie inside the Tri-View.

A single vertex can be selected or deselected with the commands EDIT SELECT INDICATED VERTEX and EDIT DESELECT INDICATED VERTEX. In either case you must indicate the vertex that you intend to change by placing the cursor beside it. Since you can do this even more easily by double clicking on the vertex, these commands are only really useful when you use the keyboard equivalents of the commands. Keyboard equivalents, where available, are listed in the menus next to the item they emulate. Additional equivalents may be created 'on the fly'. See chapter 14 for more details.

The final selection commands that we will consider at this time are EDIT SELECT INDICATED EDGE and EDIT DESELECT INDICATED EDGE. If you place the cursor on an edge, these commands will select or deselect the two vertices at the ends of the edge. This can be useful when dealing with very complicated objects when several vertices are clustered together and you do not know which one is connected to a particular edge.

There are so many ways of selecting vertices (a few more will be added in Chapters 6 and 10) because the key to manipulating complicated objects is to be able to select just those vertices that you need for a particular operation.

Erasing objects

Sculpt Animate is an editor for solid objects. So far, we have seen how to create vertices and edges, and--by the rules--make faces. What happens if a mistake is made?

The EDIT ERASE portion of the menu provides a number of options. These commands remove items from the current scene in RAM memory, but they have no effect on scenes that have been saved, unless you save over an old scene with the altered one. Be a little cautious with the erase commands, anything you erase is gone forever--unless you have saved the scene beforehand. It's a good idea to save your intermediate work at intervals.

The first selection in the ERASE submenu, EDIT ERASE SELECTED VERTICES does what you might expect. Only selected (i.e., yellow) vertices are removed. Since an edge is defined in terms of vertices, any edge that is connected to a selected vertex is also deleted, and since a face requires three edges, the removal of edges may result in faces being erased also. When you remove an object's vertices, you remove its foundation.

Note that EDIT ERASE SELECTED VERTICES works on *all* selected vertices, so you should be careful that you do not have any selected vertex outside the Tri-View, otherwise it will vanish also. It's a good idea to use EDIT DESELECT ALL before selecting vertices to erase, or at least check the status display (chapter 3) to make sure none are already selected.

The next erase command, EDIT ERASE SELECTED EDGES, is very similar. It operates on selected edges, without removing any vertices. A selected edge is one that has selected vertices at both ends. For example, if you had made a cube with diagonal braces on all six sides, you could open up one side by removing a brace. Simply select the vertices at either end of the edge to be removed. The EDIT ERASE SELECTED EDGES command will remove the edge. Again, if any edge of a face is removed, the face disappears.

BUILDING OBJECTS

In addition to being able to erase selected items, you can also erase them individually. The commands EDIT ERASE INDICATED VERTEX and EDIT ERASE INDICATED EDGE work on a single edge that is indicated by the cursor. Again, if more than one edge is near the cursor, look in another window to make sure the correct one is indicated.

The next two commands can be used to remove lamps from the scene. EDIT ERASE INDICATED LAMP will erase a single lamp. Indicate the lamp as you would anything else, by placing the cursor within a pixel or two of it in the currently selected window. The command EDIT ERASE ALL LAMPS is much less subtle, it just removes all the lamps.

The last erase command that we will consider at this time is EDIT ERASE ALL. It removes all vertices, edges and faces as well as all lamps. Because this could be catastrophic--for example if you have just built an elaborate scene and neglected to save any part of it to disk (which is not wise in any case)--Sculpt Animate will ask if you really want everything erased. This gives you a chance to retract the command.

Color

By now, you must have become aware that the objects you have made have been white, and that the effects of shading have only created different shades of gray. Every time Sculpt Animate creates a face, the face is given the current default color, whatever that happens to be. White is the default color when you run Sculpt Animate. If, at any time, you want to see or change the current default color, you need to execute the menu command EDIT MODIFY FACES.

The requester that pops up can be used for several purposes, as we shall see. This single requester lets you change every available surface attribute of the faces of your objects. It is divided into three regions, one each for these three attributes. The regions are accordingly marked

'SMOOTHING', 'TEXTURE', and 'FACE COLOR'. We will be concentrating on the 'FACE COLOR' region of this requester for now. The next two sections discuss the others.

FACE COLOR		SMOOTHING
R		OFF
G		FETCH
B		SET
H		TEXTURE DULL FETCH SET
S		
V		
FETCH SET		
BLEND		
CANCEL OK		

The Modify Faces Requester

The current color is displayed in the box near the bottom of the requester. The set of six sliders can be used to control the color--ignore everything else for now. The first three sliders control the amount of red, green and blue. To operate a slider, point at the triangular knob and hold down the left mouse button. If you move the mouse from side to side, the knob will move to follow it, and the current color will change. With a little practice, you can mix the red, green and blue to achieve almost any hue.

If you find it difficult to get just the right shade with the RGB sliders, try using the lower three that are labelled H, S and V for hue, saturation and value. It is a good technique to start by moving the S slider and the V slider all the way to

BUILDING OBJECTS

the right. If you then move the H slider very slowly it will change the color smoothly from red to green to blue and back to red again, with many shades in between.

Having established the correct hue, you can move the saturation or S slider to the left in order to dilute the color with white. Finally you can use the V slider to make the color darker or brighter.

Once you have found the correct color, click the gadget marked 'OK' at the bottom of the requester. From now on, any new faces that are generated will have that color.

The color of a face can also be changed at any time. First, you must select the vertices that surround each triangular face whose color you wish to change. Then execute the command EDIT MODIFY FACES and use the sliders to select the right color. To set the color of the selected faces, click the button marked SET that's in the FACE COLOR region of the requester. The change will take place when you click OK.

You will have to exercise some care in selecting faces to be re-colored. Consider the case of a red triangular face in the middle of a blue surface. The red face's vertices are all shared with blue faces. If you select all the blue faces, you will also select the red face! This may not be what you wanted. If this sort of situation occurs, leave one blue face adjacent to the red unselected, and re-color it later. It is often best to make differently colored faces separate, unconnected, objects.

The FACE COLOR region of the MODIFY FACES requester can also be used to examine the color of one or more faces. Select the faces that you are interested in by selecting the vertices that surround them, then invoke the modify faces requester.

If all the selected faces have the same color, then the button marked FETCH will be solid and the button marked BLEND will be shown in 'ghosted' text. If the selected

faces do not all have the same color, then the FETCH button will be ghosted and BLEND will be solid.

When the FETCH button is active, you can click on it to make the displayed color the same as that of the selected faces. The BLEND button is similar, except that it will display the average of the colors of the selected faces. Be warned that the average of several colors is not always what you might expect, it is defined by averaging the red, green and blue components separately.

Again, the displayed color will become the current color when you click OK. If you click SET in the FACE COLOR region and then click OK, the color of any selected faces is also changed.

If you call up MODIFY FACES and both the FETCH and BLEND buttons are ghosted in the FACE COLOR region, it means that no faces are selected. For a face to be selected, the vertices at all three corners of the triangle must be selected. It is easy to have many vertices selected without having any faces selected. One particularly confusing situation can arise if two vertices have been created in the same place. This can cause something that looks like a triangle to exist, and still not be a face.

Texture

The second attribute that can be set for a face is its texture. It can either be dull, shiny, mirror, luminous, glass or metal. A face's texture has no useful effect in any of the non-HAM rendering modes.

The current default texture value is shown in a box gadget in the modify faces requester, under the title TEXTURE. You can change the current default texture by clicking on the box. If you click repeatedly, the texture values cycle through their sequence. You can reverse the direction of the cycle by holding down the ALT key while clicking on the button.

BUILDING OBJECTS

A DULL texture scatters light uniformly in all directions so no highlights will be shown, although the surface will be shaded from bright to dark portions. A SHINY finish throws most of the light uniformly, but reflects some in a particular direction, which can cause an object to glint, or have 'specular' highlights. A surface with a MIRROR texture reflects all the light in a particular direction, so that reflected images may be seen in the mirrored surface. If you also set a color other than white, the reflection will also be colored.

If you color a mirrored surface, reflected colors are altered as though they had been through a color filter. This can have surprising results. For instance, there is no red light in a blue sky, so the sky will look black if it is reflected in a red mirror.

A LUMINOUS surface emits light regardless of what light shines on it. This can be useful if you are constructing laser beams, bolts of lightning or lighted buttons on a control panel. Note that luminous surfaces cannot illuminate other surfaces, but they shine into the camera so that they have the same unshaded color in the resulting image that you set in the COLOR FACES region of MODIFY FACES.

A GLASS surface reflects about 20% of the incoming light just like a mirror, while the rest passes through. Because a face represents a surface and not a solid, Sculpt Animate does not regard an object built entirely of glass as being solid, so no refraction takes place. Glass objects are useful as canopies of aircraft, or as windows in buildings. A METAL surface combines the properties of a shiny and a mirrored surface. Some light is reflected and some is scattered, giving a metallic effect.

The texture of selected faces can be changed to the current default texture by the SET button in the TEXTURE part of the modify faces requester. If all the textures of selected faces are the same, then the FETCH button will be unghosted and you can use it to change the current default texture to that of the selected faces.

MIRROR, GLASS, METAL, and especially LUMINOUS textures can create strange effects when used in the SKETCH, PAINTING, and SCANLINE PAINTING modes. Avoid using these textures if you intend to use one of these modes for your final image.

Smoothing

Smoothing is a special attribute of faces which causes one face to blend smoothly into the next, so that there is no visible edge between them. Smoothing will be especially useful when you begin building more complicated objects in the next chapter. Just as it does with color and texture, a newly created face inherits the current smoothness setting.

The current smoothness value is indicated in the modify faces requester; it is either ON or OFF. If a face is smoothed, then it will reflect light as though the surface was smoothly curved, instead of faceted. This attribute is only effective when you make images in the imaging modes that use HAM. The particular method that is used to calculate the curvature is called Phong smoothing.

If you wish to change the smoothing parameter for a face or faces that are selected, first change the current smoothing value by clicking on it to toggle it between ON and OFF, then click on the appropriate SET button.

As with texture, if all the selected faces have the same smoothness, you can use the FETCH button to change the current default value to that of the selected faces.

If you are constructing an object that is uniformly curved, like an egg, all you have to do is to turn on smoothing for all its faces. If your object combines several smooth portions that meet at an angle, like a crease, you will need some way to inform the program of your wish, or it will try to smooth out the crease.

Because the smoothing algorithm works to smooth adjacent connected faces, you can disconnect the two

BUILDING OBJECTS

surfaces that meet to form the crease. A small gap does not even matter, because you can make it small enough that it will not show in the final image.

Suppose you were constructing a solid version of the letter 'S'. You would build the front flat portion as a single connected shape, and the curved edge as a second object with smoothing turned on. If you need a rear surface, it could be built as a third object.

The only drawback of this method is that the `SELECT CONNECTED` command would have to be used three times to select the entire letter. If this is a problem, connect the rear to the side with one single edge, then connect the side to the front with another. These edges will not create faces, but they will connect the three parts together.

Flat reflective surfaces can also confuse smoothing. If a complex flat surface has been created with a `SHINY`, `MIRROR`, or `METAL` texture, then Phong smoothing may distort reflections in this surface. Flat surfaces in these textures may need to have smoothing set to `OFF`. Smoothing works very well on curved surfaces in these textures, though.

Smoothing can be a great timesaver. You will learn that objects can be created with almost any degree of detail in `Sculpt Animate`. By making a rather coarse object and giving it the attribute of smoothness, the rough edges will be evened out, so that it will be almost indistinguishable from a more finely built object. This can be a great savings in effort, memory use, and rendering time.

Keep in mind, however, that Phong smoothing can only smooth surfaces--it will not smooth outlines. This is not so bad, though, since a small amount of segmentation in an object's outline is much less noticeable than the same amount of segmentation on its surface. Sometimes it's good to use more vertices on an object's profile, leaving the rest fairly coarse. This puts the detail where it will do the most good.

The Grabber

Up to this point, we have assumed that the objects that you have built are perfect, that you have never made a mistake or changed your mind. Except for erasing all or part of an object, there has been no way to alter the objects you have created.

The Grabber is a low level tool that you can use to change the position of one or more vertices. Like most Sculpt Animate facilities, the Grabber manipulates selected vertices, and leaves unselected vertices alone.

You invoke the Grabber tool by clicking the Grabber Gadget, which resides in the lower left corner of each Tri-View window. Its icon is supposed to look like the crane operated gripping tool that is used in automobile wrecking yards. If you click on the Grabber icon with the left mouse button, you will see its jaws close. It is then active. While the Grabber is active, the normal crosshair cursor will be replaced by a copy of the Grabber Gadget icon.

The Grabber may be dropped (turned off) by clicking on its gadget a second time, or by clicking the right mouse button. As an alternative to the gadget, the Grabber tool may be activated by the menu command `TOOLS GRABBER`.

While the Grabber is active, you may move the new cursor anywhere in the scene, just like the normal cursor. The difference is that any vertices that are selected will follow along, maintaining their position relative to the cursor.

When you move a vertex, you also move or distort any edges and faces that are built upon it. For instance, if you selected and grabbed just one vertex of the cube you built a short while ago, you could move it without moving the rest of the cube. The edges and faces built upon the moved vertex would stretch to accommodate its new position. You could also select and grab the entire cube, then move it without any distortion.

BUILDING OBJECTS

Now give the grabber a try. First clear the Tri-View with an EDIT ERASE ALL command. Next make three vertices, and join them with a click on the Edge Builder Gadget. Deselect one of the vertices and turn on the Grabber. All the time that the Grabber is on, the selected vertices will move whenever the cursor is moved. The edges will stretch to follow. Notice that the cursor doesn't have to be in any particular place when you invoke the Grabber. The selected vertices will always maintain the same position relative to it when the Grabber is active. In this way large selections of vertices may be moved without altering their structure.

To try a more elaborate example, EDIT ERASE ALL to clear the Tri-View. Then PROJECT LOAD SCENE and request littlehouse. When it has loaded, deselect all the vertices with the EDIT DESELECT ALL command and move the cursor to one of the vertices at the top of the roof. Then execute the EDIT SELECT CONNECTED command and the whole roof structure will be selected. Turn on the Grabber and move the mouse pointer to the North window (the one with W, U, E and D at its edges). Then move the cursor up and raise the roof!

The grabber can also be used to move a lamp. To do this, you must first DESELECT ALL if there are any selected vertices, otherwise they would be grabbed. Next, indicate a lamp with the cursor. When the grabber is turned on, it can then be used to drag the lamp to a new location. Chapter 7 also describes this.

The Expand, Contract, Center, and Move Tri-View gadgets all work correctly when the Grabber is active, so you can move objects and lamps anywhere in the scene. You must press the right mouse button to use menu commands, so these are not available when the Grabber is active.

Avoid the Rotate (chapter 6) and Select/Deselect gadgets when the Grabber is active. Their behavior in this situation is explicitly undefined, and they can have confusing effects. At the time of this writing, the rotate gadgets will rotate the

BUILDING OBJECTS

selected objects as though the cursor had not been moved. The Select/Deselect gadget will only deselect vertices whose original (pre-grab) position is visible in the Tri-View. Deselected vertices will go back to their original position.

CHAPTER 5

POWER TOOLS

In this chapter we will present some of the 'power tools' that enable you to easily build elaborately shaped objects. The objects you create using these tools are made up of the same elements that were introduced in the last chapter--vertices, edges, and faces--so you will be able to adjust and manipulate these new structures in the same way, if necessary. In chapter 6 even more tools will be introduced that can be used to transform the basic shapes created in this chapter into almost anything.

Spheres

Because Sculpt Animate builds everything from triangular faces, it cannot represent a sphere exactly. It so happens, however, that a geodesic structure--constructed of triangular elements--can represent a good approximation of a sphere. Though it would be possible to build such a structure using the tools discussed so far, it would take extreme patience and an unusual fondness for geometry.

Fortunately, Sculpt Animate knows how to construct spheres, and other objects, for you. To have it build a sphere, execute the menu command EDIT ADD SPHERE. A requester will pop up and ask you how many times you wish to subdivide the structure. For now, simply press the OK button and Sculpt Animate will construct a 20-faced geometrical figure called an icosahedron.

To take a look at this new object, set up a lamp and make a picture of this object with the SNAPSHOT mode. It is a pretty poor approximation to a sphere, but if you modify all its faces, setting the smoothing option, a more credible image of a sphere will result.

POWER TOOLS

You have seen (in the last chapter) how to apply the smoothing attribute to an object, but we'll review the procedure. First, you must first select every vertex of the object. Since the icosahedron lies entirely within the Tri-View, you can use the Select Gadget to do this. Next execute the EDIT MODIFY FACES menu command, ensure that the current smoothing state is ON, and then click the SET button.

Now make a new image of the object. Although the outline still seems angular, the shading will make the object's surface look quite smooth. If the final size of the object is to be fairly small in the picture, this may adequately represent a sphere.

To make a better approximation to a sphere, we may subdivide each face. The EDIT DO SUBDIVIDE menu command will divide each selected edge in two, creating a new vertex at the midpoint of each edge. Then it links these midpoints with new edges. Each selected face will be divided to make four. Try this operation on the icosahedron, being sure that each vertex is selected.

The resulting object will have 80 faces, but will still look like an icosahedron, since the new faces lie in the same planes as the old ones. What we need to do is to 'inflate' the object by pushing the new vertices outwards a small amount.

There is a command EDIT SNAP CONNECTED VERTICES TO SPHERE that does just what we want. Unlike most commands, this command does not operate on selected vertices. Rather, it works on all vertices that are connected to the indicated vertex. Remember that if you want to indicate a vertex, move the cursor to it in one window. The command figures out where the center of the connected object is, and then moves all its vertices so they are the same distance from that location.

To see this happen, move the cursor to a vertex of the subdivided icosahedron, then execute EDIT SNAP CONNECTED VERTICES TO SPHERE. The resulting

object will look quite spherical, even in the Tri-View. Make a picture and see.

You could repeat this sequence of SUBDIVIDE and SNAP CONNECTED VERTICES TO SPHERE to build better and better approximations to a sphere, but such extravagant detail would be lost to the relatively low resolution of the Amiga's screen. Each subdivision will quadruple the number of faces, and before long Sculpt Animate will start to run very slowly because of the sheer complexity of the object. If you persist, you will run out of memory. A smoothed one- or two-subdivision sphere is usually more than adequate.

You might have already guessed that you can subdivide the sphere automatically by specifying the number of subdivisions when the requester for EDIT ADD SPHERE pops up. Instead of just pressing the OK button, click on the box containing the default subdivision number, erase the default with the DEL or BACKSPACE keys, then type in the desired value and hit RETURN or click OK.

Subdividing objects

The preceding section described how the command EDIT DO SUBDIVIDE may be used to increase the detail of a sphere by dividing each selected edge into two and each selected face into four.

The command may be used on other objects as well, but it's important to remember that, by itself, EDIT DO SUBDIVIDE only adds vertices and edges--not detail. Never subdivide an object unless you intend to move or alter the added vertices in some way.

For instance, if you wanted to add a nose to a potato, there might not be enough vertices in the area of interest to construct this detail. You could select just this part of the surface and subdivide it until there were enough vertices to work with. Of course, you could also construct the nose

separately and simply move it into place--there are often a number of ways to achieve a desired effect.

There is one special function that the SUBDIVIDE command will perform. If only two vertices are selected, and they are *not* joined by an edge, then the SUBDIVIDE command works quite differently. A requester pops up asking you how many vertices you want inserted. New vertices are placed evenly between the two selected ones and they are joined by edges that lie along a straight line. No faces are created, so this new object would not be visible in an image. This feature will prove helpful when you begin to create splines and motion paths in chapters 8 and 10.

Other curved objects

Besides spheres, the following curved objects can be created with the appropriate EDIT ADD command: hemisphere, disk, circle, cylinder, tube and cone. In each case you will have to enter a number to specify the degree of detail that is required.

In each case except circle and tube, a complete closed object will be built. For hemispheres you must specify a number corresponding to a number of subdivision operations, while for the other commands, the number that you give represents the number of rotation steps for a full circle.

Cubes

Angular shapes like a cube can be represented precisely by Sculpt Animate, although each rectangular face must be divided into triangles by a diagonal edge. The EDIT ADD CUBE command will construct a perfect cube in the Tri-View--complete, with a diagonal division on each side.

The cube that is constructed will be automatically sized to fit comfortably within the current size of the Tri-View. All

of the EDIT ADD commands behave this way. In the next chapter we will introduce a mechanism called EXPAND that can change the size of objects, as well as their dimensions.

With this mechanism, a simple cube may be transformed into any size or proportion of box. For now, though, we will use the Grabber Gadget to do a similar transformation. Although they are somewhat more awkward than EXPAND, the techniques discussed here can be used to create a number of more subtle effects.

Suppose we wanted to build a block that is twice as large in one direction as it is in other directions. First, clear the Tri-View with an EDIT ERASE ALL. Turn off smoothing in the EDIT MODIFY FACES requester. Then make a basic cube with EDIT ADD CUBE.

Since the cube is sized to fit the Tri-View, we need to expand the Tri-View to make room for our transformation. Click on the Tri-View Expand Gadget twice. Now we want to select the four vertices that make up one end of the cube. One way is to move the Tri-View so that part of the cube sticks outside the Tri-View. Then use the Select Gadget to select the four vertices that remain. When this is done, move the Tri-View back so you can see the whole cube.

Now look at the three windows and find one that will let you move the cube's selected side away from the unselected side. The window whose Move Tri-View gadgets you were using should do. Finally, turn on the Grabber and--in the window you just chose--pull out the end of the cube until it reaches the desired length.

If you do not have a steady hand, you may find that you have pulled the cube out to a lozenge shape. (Chapter 13 will tell you how to avoid this.) Try to remember this shape--and any other shape you create by accident. You never know when you might find a use for it!

POWER TOOLS

Prisms

The command EDIT ADD PRISM works just like the command for cubes. The prism will be oriented as though it were the roof of a building, but in the next chapter you will see how to tilt and rotate it into any position.

Just as with the cube, you can easily alter this basic shape to achieve any number of effects. For instance, if you want to create a steeply-pitched roof, you can select the two upper vertices by double-clicking them, then use the Grabber to raise the ridge. You can also move the ridge side to side, creating interesting asymmetrical effects. Similarly, if you want anything other than a square plan, turn off the Grabber and select just one end of the roof, then grab and move that.

Again, subsequent chapters will introduce even quicker ways to alter basic shapes like the prism.

Drawing curves

So far, the shapes we have constructed have had simple, regular profiles; either circular, square, or triangular. The majority of objects in the real world are not so regular, however. Most are clearly distinguishable by a unique profile.

Sculpt Animate has several features which allow you to take one or more profile curves and create solid objects from them. In Sculpt Animate, these curves are approximated by a series of vertices joined by edges. Although we could make them using the methods of chapter 4, there is an easier way.

To see, execute the command TOOLS CURVE. Notice that the form of the cursor has changed: you have picked up the curve tool. This tool works very much like the regular cursor. You can click or hold down the left mouse button to move the curve tool around. If you press the right button while the left button is held down, a new vertex is created--just as before.

Unlike the regular cursor, though, subsequent vertices will be joined to their predecessor with an edge. This lets you easily create a linked set of vertices. As you move the curve tool, the last vertex will be linked to the cursor by a dotted blue 'rubber band line'. This reminds you that you are about to create an edge, and shows you the orientation of the edge.

While the curve tool is active, the gadgets for moving and sizing the Tri-View may still be used to move around the scene.

You may release the curve tool at any time by clicking the right mouse button. If you position the tool very close to an existing vertex, the curve will be closed by creating an edge to the vertex, and the curve tool will be dropped. In this way you can create closed loops.

The Fill command

A closed loop would be just what you needed if you had to create a flat object with an arbitrary shape, like a letter on a billboard. If you tried to make an image of such an object, though, it would be invisible, because although you would have created vertices and edges, there would be no triangular faces. You could construct additional edges to fill the interior of the loop, but Sculpt Animate provides a command to do this for you--EDIT DO FILL.

To try out this command, construct a closed loop in some interesting shape and ensure that all the vertices are selected. Then execute EDIT DO FILL. Vertices will be constructed within the loop to fill its outline with faces.

Erase the object and construct another closed loop. Then choose the curve tool again and make another loop entirely inside the first. If you use the EDIT DO FILL command, this time the second loop will serve to define a hole in the otherwise filled object.

POWER TOOLS

Of course, if the inner loop is triangular, it will be a face itself. If you wish to create a triangular loop that is not a face, just subdivide one edge of a triangle, then erase the interior edge that will be created.

The rules for using the EDIT DO FILL command are as follows. Only selected vertices are considered. They must all lie in (or nearly in) a single plane. The plane can be oblique--that is, it does not have to be parallel to one of the sides of the Tri-View. The selected vertices and edges must describe one or more closed loops. Finally, no two selected edges may cross.

The Edge Maker Tool

If you have experimented with the Fill command, you may have noticed two things. For complicated shapes, the command seems to execute slowly. You might even be able to do the job more quickly yourself. You may also be surprised at how the Fill command chooses to add new edges.

In general, rendering images will be faster if you avoid long skinny triangles. The fill command sometimes makes such skinny triangular faces, even though you could probably figure out how to fill the loop more intelligently. The Fill operation is one of those things that humans are good at, but that computer programs are apt to perform less well.

If you choose to do the Fill operation manually, there is a tool that makes the job much easier. To try it, first make a closed loop--it doesn't matter if the vertices are selected or not. Next, call the command **TOOLS EDGE MAKER**. Notice that the cursor changes to a shape that may remind you of the Edge Builder Gadget.

To use this new tool, move the pointer to a vertex, then hold down the left mouse button and click the right button once. This starts the edge construction. Move the cursor and notice that a rubber band line goes from the vertex to the

cursor. Move the cursor to a second vertex and do the same thing again--hold left, click right. An edge will be made between the two vertices.

You can now point to a new vertex and click the right button to define the start of another edge, or click on the right button without holding down the left button in order to drop the tool.

Spinning new objects

Many objects have symmetry about an axis. Examples include a clarinet, a doughnut, a spindle, a chess pawn, a candlestick, and an hourglass. All of these objects can be described as a curve which has been swept about an axis. The EDIT DO SPIN command is the mechanism that Sculpt Animate provides for generating these shapes.

To make this command work, you must first create a curve that describes the outline of your object. This is simple enough if you use the tools that have already been described.

Once you have designed your pattern, make sure no other vertices are selected in the scene (with EDIT DESELECT ALL), then select the pattern curve with EDIT SELECT CONNECTED.

Next you must indicate the axis about which the pattern is to be spun. Do this by placing the cursor some distance above or below the pattern curve in one of the Tri-View windows. It is assumed that the axis runs horizontally across the screen, passing through the cursor in the selected window.

When you are satisfied with the curve and the location of the axis, execute the EDIT DO SPIN menu command to generate a new shape. The command will display a requester asking how many steps are to be taken as the curve is swept around the axis. You must use at least three. The more you choose, the smoother will be the resulting object.

POWER TOOLS

But more steps will also consume more memory, having a similar effect to using many subdivisions on a sphere. The default value of 12 is usually adequate.

After you have selected the number of steps, you will be asked to provide a spin angle. Normally you should accept the default angle of 360 degrees, in order to produce a full surface of revolution. If you decide to use an angle less than 360 degrees, then a partial spin will be performed--just the thing to model a slice of cake.

Experiment with the spin tool to see how many different shapes you can build. Try using a semicircular pattern to make a sphere. How does it differ from a sphere made with the EDIT ADD SPHERE command?

Note that when you spin a curve less than 360 degrees, you only construct the curved part of the object. In the case of a slice of cake, the flat parts that would be cut with a knife are not filled. You would have to select the outline of these parts and use the Fill command to complete the object.

You will probably want to use SMOOTHING on objects created with the SPIN command, just as for spheres. This allows you to specify fewer steps of revolution for better efficiency, while still creating a realistically curved surface.

Helices

A variation of the spin command is provided that lets you build objects with helical or spiral symmetry, for example a screw thread, a catherine wheel or a coil spring.

As we've seen, the spin command lets you choose a profile curve and spin it about an axis by a specified angle. The command EDIT DO MAKE HELIX lets you do this while at the same time letting the profile traverse along the axis, move away from the axis, or both.

Somehow you have to tell the program how you want the profile to be moved as the rotation is performed. This is

done by means of a control curve. The control curve is made of a series of linked edges and vertices, in the same way as profile curve. Each edge of the control curve corresponds to one complete revolution, or twist, of the helix. The length of each edge of the control curve will determine the length of the corresponding twist.

This is much easier to demonstrate than to explain, so go to your machine and clear the Tri-View with an EDIT ERASE ALL. Create a single vertex near the top left corner of the 'down' window. The new vertex is selected--leave it that way, since it is to be the profile curve and so must be selected. Do not worry that the profile is a single vertex and not really a curve at all, it will do fine for now.

Now select the CURVE tool from the TOOLS menu and go to the bottom of the same window. Create a two-vertex curve so that both vertices are on the bottom of the window, spaced apart by about one quarter of the window's width. These two vertices and their edge will be our control curve.

To indicate that this is the control curve, place the cursor on the vertex at one end of the edge. Either vertex could be used, but choose the leftmost one for now. It does not matter if the vertices that make up the control curve are selected or not, they will only be used as a measure for moving the profile. Any other selected edges and vertices will be used to construct the object.

As with the SPIN command, the axis of revolution for the helix is taken to be a horizontal line in the current window, passing through the cursor. Since the cursor is busy indicating the first vertex of the control curve, you must naturally construct your control curve so that its first vertex lies on the desired axis.

With the cursor positioned at one end of the control curve, execute the menu command EDIT DO MAKE HELIX. You will be asked how many steps you wish to be used for each turn. For now, accept the default value of 12 by clicking on OK. Next you will be asked the total number of steps that you need. Again, use the default value of 48 by

POWER TOOLS

clicking on OK. This means that you will produce a four turn helix, with 12 steps per turn.

If you have done everything correctly you will see a few inverted V-shaped objects in the down window, some diagonal lines in the north window, and part of a circle in the west window. To see what has really happened you will have to expand the Tri-View a few times. Then all should become clear. The object that you have created is a simple helix, the same shape as the wire springs you can find inside retractable ball point pens.

To get a better look at your creation, you can render an image of it. Set the rendering mode to OBSERVER WIRE FRAME (you have not created any faces, so the other modes will not reveal anything) and select a FULL image size. Position the target in the middle of the helix, and move the observer off to one side. Use OBSERVER START to make an image. Try this again a few times, moving the observer around until you are sure of the shape that you have created.

Clear the Tri-View again and repeat the example, except this time make the control curve a vertical rather than a horizontal edge. Remember to place the cursor on the end of the control curve. Instead of a helix, this time the command EDIT DO MAKE HELIX will build a spiral curve.

The control curve can be of any length, but it does not make any sense to let it have more edges than the total number of turns of the helix. Clear the Tri-View and repeat the first example, but this time use five vertices, creating a four edge control curve. Place each vertex of the control curve on the bottom of the window, but space them out so that each edge is longer than the previous one. Construct a helix and note that the helix becomes more stretched with each turn. See how the pitch of the helix matches the spacing of vertices along the control curve. Experiment with control curves that are neither horizontal nor vertical and see how the helix develops spiral tendencies.

Now you are ready to build real helical objects. Clear the Tri-View and use the curve tool to draw a small freehand

circle with about half a dozen vertices near the top left corner of the down window. Drop the curve tool, then call it up a second time to create a horizontal control curve with two vertices, like the first one you did. Move the cursor to the leftmost control vertex. Ensure that the length of the control is larger than the diameter of the freehand circle, and make sure the circle is all selected. Now select EDIT DO MAKE HELIX again.

You have created quite a complicated object, much like the coil spring in the front suspension of some automobiles. The helix is made up of faces this time, so you can place a lamp and use some imaging mode besides WIRE FRAME to look at it.

Although you can build a multitude of open helical objects with this command, there is one special way that it can be used in order to create helical surfaces. A helical surface is one where one turn is connected to the next, with no space in between. In order for this to work, there are two additional rules that must be satisfied before executing the helix command.

The first rule is that there must be no control curve, so the cursor must not be near any vertex. The cursor is still used to indicate the axis of rotation, though.

The second rule is that the profile curve must be a simple curve. That is, it may have no branches, and it cannot close up on itself in a loop.

As an example, clear the Tri-View and construct a small V-shaped object at the top left of the down window, comprising three vertices and two edges. Move the cursor to the bottom of the window and execute the command to make a helix. The role of the control curve is now played by the profile curve, except that the overall size of the profile sets the pitch and change in radius. You will see that the V-shape creates a surface like the threaded portion of a bolt.

If one leg of the V were higher than the other, then the threaded surface would taper. As you might expect, if the

POWER TOOLS

previous example were repeated with a V-shaped object that was drawn on its side, then a surface with a helical groove would be created, like the shape of the groove in a phonograph record.

Extruding shapes

The last outline-transforming tool that we'll be examining in this chapter is the extrude tool. This tool lets us use a curved pattern like a cookie cutter, to make solid shapes from a two-dimensional profile.

Once again the CURVE tool is used to draw one or more curves or closed loops with selected vertices. When you are satisfied that you have created the cross section of the shape that you want to extrude, execute the **TOOLS EXTRUDE** command. There will be a short pause, during which the grinding gear image will replace your mouse pointer.

When your normal pointer returns, notice that the Grabber tool has been turned on. Now select a Tri-View window where the curves can be viewed edge-on. If you move the cursor in this window, you will see the original patterns divide into two. An unselected (purple) copy of the original pattern remains fixed, while the selected (yellow) copy can be dragged with the cursor. New edges appear, joining the two outlines. These edges have been created by the extrude tool. When you are satisfied with the extruded shape's form, click the Grabber icon to turn it off.

If you had traced out the outline of a building in plan form, you could use the extrude tool to raise the walls with one stroke of the mouse. It can also be used to build three dimensional alphabetic characters, any number of stamped or extruded forms, and many other objects.

One useful property of the extrude tool is that if you use it on a curve that has been filled, the interior edges do not extrude to form interior partitions. This prevents useless faces from being created within extruded objects, where they would waste memory and processor time.

CHAPTER 6

MORE TOOLS

In this chapter we will look at more ways that we can manipulate and build objects.

Saving and loading objects

If you build an object, then you can save it to disk with the PROJECT SAVE SCENE command. A file requester will pop up to enable you to specify the name that you wish to use for the file. This requester was described in detail in chapter 2. It lets you move around among the volumes and drawers in your Amiga's file system. When you have found the right drawer, you may either select a pre-existing filename or type in a new one in the provided box. If you pick a name for saving that has already been used, the old file will be destroyed. You will be warned before this happens, though, and allowed to cancel the save if it is not what you want.

A scene is made up of four parts: lamps, the observer, the world, and objects. When you have finished picking a filename to save the scene under, another requester will appear, listing these parts. Next to each part's entry is a 'toggle gadget' which may be switched from YES to NO and back just by clicking within its border. Items marked 'YES' will be saved, items marked 'NO' will not.

'Lamps' is the top item in the list. It simply specifies whether lamps will be saved. You may not want to save the lamps in a scene if you are simply saving its objects to be used in another scene. On the other hand, you might have a particularly pleasing arrangement of lamps that you wish to save for use in other scenes.

'Observer' is the next item. If its gadget is set to NO, the observer's location and target will not be saved. Neither

MORE TOOLS

will any of the parameters associated with the observer menu, such as rendering mode and image size and type. These things are saved with a scene in the Observer part.

The 'World' is described in more detail in the next chapter. It includes details about the sky, ground, and background illumination for a scene.

'Objects' is the final and most important entry. The objects in a scene are its vertices, edges and faces. Color, texture and smoothness are also stored as part of the objects. If this item is set to 'NO', no objects will be saved at all. If it is set to 'YES', they will be saved according to the gadget marked 'Which?' at the bottom of this requester.

The 'Which?' gadget is usually set to 'ALL', which means that all objects will be saved if 'Objects' is set to YES. If the gadget is set to 'SELECTED', however, only those objects in the scene that are selected (with yellow vertices) will be saved.

By carefully selecting which parts of a scene to save or load, you can combine the best parts of different scenes--lamps from one, objects from another, etc. Be careful, though! If you leave some of these items set to 'NO', then the next time you save a scene you might not save all you wanted to.

PROJECT LOAD SCENE behaves very much like PROJECT SAVE SCENE, except, of course, that it loads the scene data back into the computer. LOAD SCENE also allows you to choose which parts of the scene to load. The selection states of objects are not stored, of course, so all objects that were saved in a scene file will be loaded in, but you may choose whether the objects that are loaded will be selected or not.

The LOAD SCENE command does **not** erase any objects that are already present in the Sculpt universe. This allows you to load in different parts from different scenes, as we have seen. If you want to clear the slate and start over

with a scene you've loaded, do an EDIT ERASE ALL before re-loading the scene

If you only need to save or load objects, then you may use the commands PROJECT SAVE OBJECT and PROJECT LOAD OBJECT. These operate a little differently from the commands that work with complete scenes. You are not even given the choice of saving or loading lamps, the observer, or the world, and only selected objects are saved.

There's a more subtle difference, as well. LOAD SCENE will always load objects into the same 'absolute' position in the Sculpt universe where they were when SAVE SCENE stored them. On the other hand, SAVE OBJECT records the position of the saved vertices relative to the cursor. When the objects are loaded in by LOAD OBJECT, they will be placed the same way relative to the cursor's location, no matter where it is in the scene.

Suppose that you had constructed a model of a grand piano and all the vertices were selected. Then you could position the cursor under one leg and save the object to a disk file. If at another time, you wished to introduce a grand piano into the Tri-View, you could move the cursor to the position where you wanted the leg of the piano to be and execute the PROJECT LOAD OBJECT command.

If you need numerous copies of the same object, you can also use the command PROJECT LOAD OBJECT to add multiple copies.

Rotating objects

We may not always want our objects to be lined up with the axes of the Tri-View. If we are building a model of a city, for instance, and we want to use the EDIT ADD CUBE command to produce buildings, we have yet to see an easy way to produce a cube that faces to the southeast, or any other 'off axis' direction.

MORE TOOLS

The Rotate Gadgets are just such a way. They may be used to rotate any selected object. To see how they work, do an ERASE ALL and make a cube. Position the cursor at the center of the cube and select all eight of the vertices (the Tri-View's Select gadget should do nicely for this). If you now click the left mouse button on one of the Rotate Gadgets at the top left border of the Tri-View window, you will see the cube rotate 5 degrees for each click. If you hold down the right Amiga key as you click, each rotation step is increased to 45 degrees. Similarly if the left Amiga key is held down, the rotation step is only one degree.

The left and right ALT keys can be used to modify the rotation gadget even more. If the gadget is used with the left ALT key, the rotation is only 0.1 degrees. This is too small to see, but, as we will see in Chapter 13, there are ways of measuring angles with great precision. The right ALT key increases the rotation angle to 180 degrees, so this is a quick way to turn an object about-face.

If you move to another window, you will be able to revolve the cube around a different axis.

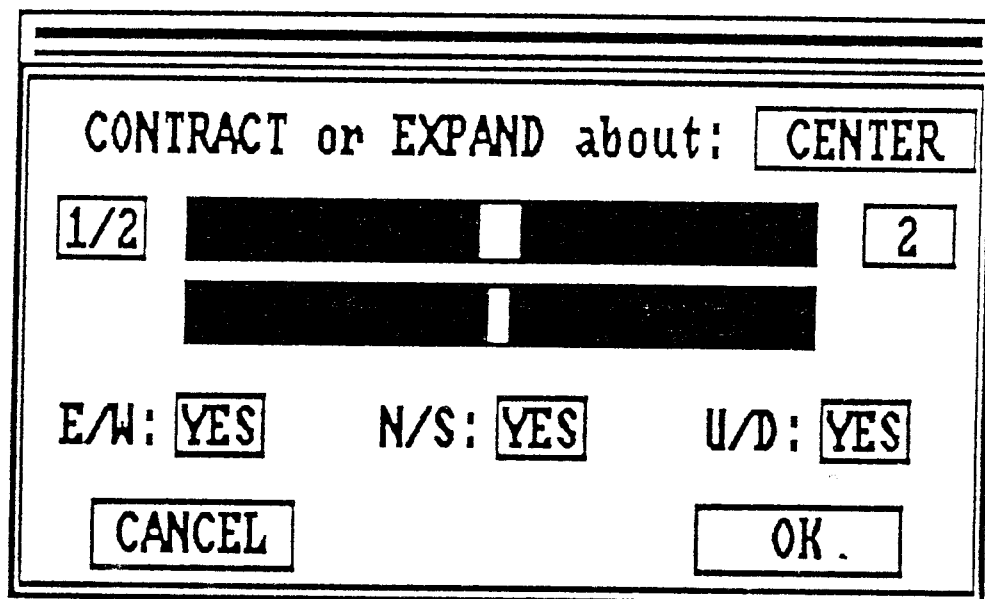
To see a different use for the rotate gadgets, expand the Tri-View to about ten times its original size and position the cursor at some distance from the diminutive cube. Click the Rotate Gadget and you will see the little cube move in a circular orbit around the cursor.

In summary, the Rotate Gadget rotates all the selected vertices around the cursor. Be careful that you do not have any vertices accidentally selected outside the Tri-View, because they will be rotated also.

For now, you will probably want to select all the connected vertices of an object to be rotated. Otherwise only part of the object will be rotated and a tangled mess can result. As you begin to create more detailed objects, though, you may find it convenient to select portions of an object's surface and alter its shape by clever manipulation of the cursor and rotate gadgets.

Expansion and contraction

Each of the EDIT ADD sequence of commands creates an object just a little smaller than the size of the Tri-View. This wouldn't do if we wanted an object of a certain size, since we can't specify the size of the Tri-View precisely. Besides, if we found that we had created an object of the wrong size, our only recourse would be to erase it and try again, or start fiddling around with the Grabber. That would hardly be practical for a pseudo-sphere with 320 faces. Fortunately there is a command that enables you to change the size of an object with great precision.



The Expand Requester

Let's learn by doing. First, make a one-subdivision sphere and select all its vertices, then execute an EDIT DO EXPAND command. An expansion control requester will pop up. Unlike some other requesters, this one has a drag bar along the top. Move the requester so that you have a good view of each of the Tri-View windows.

The EXPAND requester has two slider gadgets across its width. Moving either slider to the left or right will move the selected vertices towards or away from the chosen expansion point. The vertices move proportionally, more or less so according to their initial distance, causing a smooth change

MORE TOOLS

in the size of the selected object. The upper of the two sliders provides the coarser adjustment. The lower slider moves the vertices very slightly. Chapter 13 introduces a measuring device that can be used to monitor the size of an object precisely as you change it. This makes the fine adjustment more useful.

There are two other sizing gadgets, one at either end of the sliders. One is marked '1/2', the other is marked '2'. Clicking these gadgets will make the selected object either half- or double-sized. If you make an object much larger, it will move outside the Tri-View, so give yourself enough room to work before you call up the EXPAND requester.

At the top of the requester is a gadget that starts out labeled CENTER. In this state, the gadget indicates that the center of expansion is at the center of the object (as defined on the next page). If you click the gadget, it will switch to CURSOR, and then to CENTROID. This lets you choose what part of the object will remain fixed as the rest expands.

You can use the gadgets near the bottom of the requester to limit the expansion to one or two dimensions. Click on the boxes and see them toggle between YES and NO. Turn off two of the directions and then try contracting the sphere. You will see it flatten out like a pancake.

This is an important principle. EXPAND can be made to work in any combination of the three dimensions. For instance, a simple cube can not only be lengthened, it can be resized in any dimension to make any kind of rectangular block. The compass marks on the Tri-View windows can be helpful in choosing which dimensions you want to resize.

Even more variety can be achieved by selecting only portions of an object to be resized. By selecting the top four vertices of a cube and shrinking them, you can create a trapezoidal pedestal (a pyramid with its top cut off). You can select individual vertices of a sphere and resize them, creating a number of spikes or dimples on the sphere's surface. Starting with a few simple objects and some imagination, you can quickly create any number of shapes.

Note that expanding relative to the cursor gives you some special options. Placing the cursor away from the its center will make the object move, as well as change shape. This makes it possible to fit an object into a given space. The CENTROID setting is similar in its effect to CENTER. It is provided primarily for completeness, but it may come in handy for some applications.

Finding the center of an object

There are times when it is desirable to have a quick and precise way to find an object's center. The command EDIT SNAP CURSOR TO CENTER can be used to do just that. It moves the cursor to the center of the selected vertices. For our purposes, the center of a group of vertices is considered to be the center of a rectangular box that is just big enough to hold all the vertices. The faces of this imaginary box are parallel to the Tri-View sides. This may sound a bit more complicated than necessary, but there are many ways to define the 'center' of an object. This definition will usually give us the location we want.

The command EDIT SNAP CURSOR TO CENTROID is sometimes useful as an alternative way to find the middle of a group of selected vertices. The centroid is the 'center of gravity' of the vertices, assuming that each vertex has the same weight. The centroid will often differ from the center, as defined above, but not always. For example if most of the vertices lie on the left side of an object, then the centroid will be offset from the center in that direction. For some shapes, like a subdivided sphere, the center and the centroid are the same, while for others, like the hemisphere, they are quite different.

Manipulating shapes

If you were to EXPAND a sphere along one dimensional axis, you might expect that it would lengthen out into an ellipsoid (oval solid). This is indeed the case. The Grabber

MORE TOOLS

Tool can also be used to perform an EXPAND function of sorts, by selectively lengthening middle sections of an object. This distinction will come in handy over time, as you build more diverse shapes.

Suppose that you wish to create a long cylinder with spherical ends. First construct a sphere with the EDIT ADD SPHERE command. Use a subdivision value of one or two. Then move the Tri-View so that exactly half of the sphere is sticking out of the Tri-View. Now if you use the Select Gadget, you will find that you have selected all the vertices on one hemisphere. Click the Expand Tri-View gadget to make room for what will follow. Turn on the Grabber and you will find that you can pull the two halves of the sphere to make the cylinder with rounded ends.

With tools such as EXPAND and the Grabber, you can build a great variety of objects just by altering pre-existing ones.

The magnet tools

If you need to pull or push several vertices, then the Grabber tool will do that for you. However it moves them all by the same amount. Sometimes you may want to move some further than others. This is what the magnet tools do.

Let's try an example. Erase all objects and create a few vertices. Ensure that they are selected. Now execute the menu command TOOLS MAGNET ATTRACT and you will see the cursor turn into an icon for a horse-shoe magnet. As usual, if you hold down the left mouse button you can drag the tool around. If, while holding down the left button, you click the right mouse button, then all the selected vertices will move a small distance towards the magnet. If a vertex is far from the magnet, the vertex will only move a small distance. Just like a real magnet, the attraction gets weaker at increasing distances. When a vertex gets very close to the magnet, the attraction gets weaker again in order to prevent the vertex from overshooting the magnet. Try it and see how the vertices move.

The strength of the magnet can be varied. If the left Amiga key is held down (as well as the left mouse button), when the right button is clicked, the strength is reduced. If the right Amiga key is held down, the strength is increased. If you select TOOLS MAGNET REPEL, everything works as before, except that the selected vertices move away from the magnet. The magnet tool is released by clicking the right mouse button without holding down the left button.

Erase everything and make three vertices arranged in a large equal sided triangle. Join the vertices with edges. Ensure that the three vertices are selected and execute the EDIT DO SUBDIVIDE command a few times until you have made a triangular grid. Position the cursor at the center of the grid, and then move to another Tri-View window. Move the cursor away from the grid and select the MAGNET ATTRACT tool. Operate the magnet a few times and see what happens. The net of points will deform and a spike will be pulled out towards the magnet.

If you repeat the experiment with a new grid, but use the magnet to repel the vertices, the grid is deformed, but the bump is rounded rather than spiked. By moving the magnet between pulses and by switching from attraction to repulsion, any number of unusual shapes can be built.

If you operate the magnet in a Tri-View that is large, the strength of the magnet is increased, so that its effect appears to be the same. If the Tri-View is contracted, the magnet gets weaker and it can be used to make very delicate modifications to a small portion of an object.

There will be a tendency for a selected object to grow, overall, as it is deformed by MAGNET REPEL, and shrink as it is shaped by MAGNET ATTRACT. Don't worry too much about this as you shape a set of vertices, because you can always use EDIT DO EXPAND to resize the resulting shape to fit your scene.

MORE TOOLS

The Selector tools

Many of the tools, commands and gadgets only work with selected vertices, edges or faces. We have already seen a number of ways to select and deselect vertices, but this function is so important that additional ways to perform it will always be welcome. It would be nice, for instance, to select small groups of vertices in a slightly more wholesale fashion than double-clicking allows.

There are two tools designed specifically for selecting and deselecting vertices in this way. If you issue the command `TOOLS SELECTOR`, the cursor changes into a yellow square. This is the Selector Tool. It represents a cube that you can move around in three dimensions. If you pass the cube over a vertex it will become selected. If this does not seem to happen, it is because you are at the wrong 'depth' in that window. Remember that you can only locate a point in two dimensions in any one window. Look in one of the other Tri-View windows to see where the cube is really located.

You will quickly learn how to drag the Selector Tool around in the Tri-View. Just as the Tri-View is a box-shaped 3D window that may be moved around in a larger 3D space, the Selector Tool is a smaller box that moves around within the Tri-View. Any vertex that passes through this box becomes selected.

You will notice that the Selector Tool is always the same size in the Tri-View windows, just like the other tool cursors. By zooming in and out with the Contract and Expand Tri-View gadgets, you make your objects larger and smaller relative to the Selector tool, so it may be used to select individual vertices or large parts of your objects. When you have finished with the Selector Tool, click the right mouse button to drop it.

While selection of vertices is very important, there are likely to be times when you want to deselect one or several vertices among many that are selected. A tool to perform this complementary function can be selected with the

command **TOOLS DESELECTOR**. This time the little cube is purple and it deselects everything that passes through it.

Duplicating Objects

Sometimes you will need several copies of the same object. The command **EDIT ADD DUPLICATE**, will cause every selected vertex, edge and face to be duplicated. The duplicate is produced at exactly the same location as the original, so it won't be visible. If you use the Grabber to move the original version (since it is selected), then the copy will be revealed. The copy is created in an unselected state. You can use this command like a brush in a paint program, to scatter duplicate objects wherever you please.

Of course, every additional copy of an object puts an additional load upon memory and computation. If you plan to create several copies of an object, you will probably want to make your prototype with as few vertices as possible. Use smoothing, not vertices, to round out curved objects, and remember that excessive detail will usually be lost to the screen's resolution.

Building mirror images

Some objects, like a gloves, come in pairs. As in the case of a pair of gloves, paired objects are often mirror images of each other. If you had painstakingly built a model of a left glove, you wouldn't want to build its mate by hand, and you don't have to. Sculpt Animate's **REFLECT** command will build the mirror image of an object automatically.

To use this facility, you must select all the vertices of the object to be reflected. Choose one window and position the cursor in it, to the left of all the selected vertices. The imaginary plane of reflection will be vertical in the current window and it will pass through the cursor position.

MORE TOOLS

Execute the command EDIT DO REFLECT, and a mirror image will appear to the left of the cursor.

You may place the cursor, and hence the plane of reflection, at any position in the window, but if a selected vertex lies on the plane of reflection (or within a few pixels of it), then that vertex will not be duplicated. Rather, it will be shared between the object and its reflection. This provides an easy way to construct closed objects that have a plane of symmetry.

For example, a human head is approximately symmetrical. If you were to build the left half of a head, you would only need to construct one eye, one ear and half a nose and half a mouth. You would have to take care that all the vertices that are supposed to lie on the center plane do so exactly. To check this, see that they lie in a straight line in one of the windows. If you select this window and position the cursor on this line, then the reflect operation will construct a complete head, with the halves 'stitched' together.

Multiple use of the extrude tool

The extrude tool was described in the last chapter, but it can be used in conjunction with other tools to build many different shapes besides simple cookie cutter slabs.

Suppose you wished to build a cylinder with a rounded 45 degree bend, perhaps a model for some plumbing fixture. Before you do anything else, clear the Tri-View with EDIT ERASE ALL. The first construction step is to click on the north window and construct a circle using EDIT ADD CIRCLE. Now move to the west window and expand the Tri-View about three times, then select all the vertices. Move the cursor to the bottom left corner of the west window and execute the menu command TOOLS EXTRUDE. Move the cursor straight across the west window, to the middle of its bottom edge. This will drag out an extrusion of the circle, making a cylinder. Press the right mouse button to disengage the Grabber.

Move the Tri-View to the north so that the selected vertices lie near the left side of the west window, three clicks on the move Tri-View gadget should be about right. Now position the cursor directly below the selected vertices at a distance about equal to half the diameter of the cylinder. This point will be the center of curvature of the curved portion of the bend. Execute the **TOOLS EXTRUDE** command and immediately click on the clockwise rotation gadget in the west window. Repeat this sequence of extrusion and rotation eight more times and you will have completed a uniform 45 degree radiused bend. Now all that remains is to extend the pipe at the end of the bend. You could use the extrude tool again and extend it in a freehand manner.

A more accurate way to extend the pipe would be to use the edge of the Tri-View as a guide, but to do that we must first rotate the whole object by 45 degrees. Make sure the Grabber is turned off and click on the clockwise rotation gadget while holding down the right Amiga key. Only some of the pipe's vertices are selected, though, so its edges will be tangled horribly. Don't worry, though, we have made an intentional mistake to illustrate something. All is not lost, you just have to rotate the unselected vertices and all will be correct. To do this, select the command **EDIT SELECT SWAP**, then you can rotate the rest of the object clockwise by 45 degrees in the same way, holding down the right Amiga key.

Now that all's well, use **EDIT SELECT SWAP** to switch the selection back to the end of the pipe. The selected end should now be lying in the horizontal plane. To complete the object the pipe must be extruded downwards. To do this while using the edge of the Tri-View as a guide, place the cursor on the right side of the west window near the top and execute the command **TOOLS EXTRUDE**. Drag the cursor down the right hand edge of the window and then turn off the grabber. Finally, select all and use **MODIFY FACES** to invoke smoothing, call up a lamp, position the observer and target, and create a snapshot of your masterpiece!

MORE TOOLS

The unslice operation

Using the tools that we have described, there is hardly any regular shape that you cannot build with precision and ease. The world is full of irregular shapes, though. Unless you wish to construct them one vertex at a time, you need a way to build them more easily. One very useful method, that's often used in the 'real' world, is to join together a number of 'formers' that describe an object's outline at various locations along its length, as though stacking sliced bread. Common examples are airplane fuselages, canoes, and the human ribcage.

In the Sculpt universe, it's fairly easy to build such formers. They can be constructed in 2D in a single window, using the Curve Tool, etc. Once you have the formers built, however, they need to be joined together. Sculpt Animate provides the Unslice operation to let you assemble such 'slices' into a solid shape.

For the command to work, each slice must be a single closed loop and all the slices must lie in parallel planes.

To achieve this, construct your first slice in a single window, using the CURVE tool. After each loop is drawn, go to another window and move the cursor to the position of the next slice. Then go back to the first window to draw the next loop. It is important to get the correct spacing between slices, as well as getting the correct shape for each loop.

If you resize the Tri-View windows so that the view you work in is large and the other two are very narrow, you can easily concentrate on a single slice. You may want to expand the Tri-View every now and then, though, and look down the length of your object to see that all the formers are lined up correctly.

As always, save your work periodically. Once you have completed a set of slices, save them in a special file before invoking the Unslice command. Keep this file for later. You will usually find it easier to edit the slices, if necessary, than to alter the completed object.

Make sure all the vertices that make up the slices are selected, and that no other vertices are selected. Execute the command `EDIT DO UNSLICE`. After a period of calculation, the loops will be joined together by edges, creating a single surface. If the end slices are triangles, then the surface will be closed.

There is no unique way to link the loops, so the `unslice` command uses a heuristic approach to tackle the problem. Do not be surprised if the solid that it generates differs from the one you anticipated. Adding additional slices, so that one slice is not too different from the next, will serve to define the shape with more precision.

The `REFLECT` operation can often be paired with the `UNSLICE` operation. Sometimes the `REFLECT` will be performed first, and at other times it is best to do the `UNSLICE` first. In either case the idea is to create either a half set of loops (like half a loaf of bread) or a set of half loops (a loaf cut down the middle). Again, save these curves before constructing the full object. You will probably wish to make changes after you have seen the complete object, and the bare profiles are valuable for this.

Hiding objects in the Tri-View

You may have already created some scenes in `Sculpt Animate` that tax the speed of the Amiga's processor. With complex scenes, it can take several seconds to refresh the Tri-View whenever you perform an operation. This aside, the Tri-View can simply become cluttered with a lot of detail that you may not need at a given time. There will likely be times when you wish parts of the scene could just be told to 'go away' for a while. Fortunately, you can do this very thing. The command `EDIT DO HIDE SELECTED VERTICES` will cause all selected vertices to disappear from view.

Hidden vertices will not take part in any Tri-View operation, and do not require any processor time when the screen is refreshed. Hidden vertices, do remain in memory,

MORE TOOLS

however, occupying RAM. You may hide part of your scene at one time, then later hide some more. Any edges or faces that are built upon a hidden vertex will also be hidden.

When you save a scene, hidden objects will be saved if the 'Which?' item in the 'Save what?' requester is set to ALL. If this item is set to SELECTED, or if you use SAVE OBJECT, the hidden vertices will not be saved. This is because hidden vertices cannot be selected.

When you want to get the hidden vertices back, use the command EDIT DO REVEAL HIDDEN VERTICES.

Creating more complicated objects

So far, you have created a number of different objects, and seen how even more objects can be created. Still, we have been dealing with relatively simple shapes. There is a reason for this: the majority of complex objects in the real world are actually collections of simple shapes. This may seem obvious, but it deserves special comment because it makes life so much easier. It's better to build a house, for instance, by creating the parts individually and putting them together, than it is to construct it all in one piece. The key to successfully modelling a complex object is breaking it down into constituent parts that can be easily dealt with.

You might wonder just how to 'put the parts together'. This depends on what you will be using the completed object for. It would be rather difficult, and in most cases unnecessary, to join individual parts of an object 'physically'--trying to make them into a single piece by carefully constructing interconnecting faces and edges. If two objects are close enough together in the scene that you can't see a separation in the rendered image, they are 'connected' well enough.

If you prefer, it's usually all right to let the faces of one part intersect with those of another, so long as it does no harm for a part to protrude through on the other side. Sculpt Animate will sort out intersecting faces for you.

On the other hand, there are some good reasons to tie certain objects together more securely. For instance, you may want to be able to select the whole object with a single `SELECT CONNECTED` command. To connect parts in this way, just use the `EDGE MAKER` tool or gadget create a single edge, tying the two objects together. A single edge between two otherwise unconnected objects cannot make a face, so the new edge will not be visible except in `WIRE FRAME` mode.

In chapter 9, you will learn a way to combine objects 'logically', giving parts of objects names combined in a hierarchy.

As you design models, you should take care that you choose an appropriate level of complexity. In `Sculpt Animate`, it is possible to model objects to a ridiculous level of detail. Given enough RAM, you could put every scupper on the hull of an aircraft carrier, or place every scale on the wings of a butterfly, but such detail would never be seen in the finished product. To include it wastes time in rendering as well as construction, and may make the scene too large to render in the available RAM.

As you look at an object you want to model, think of each part in terms of how you could create it. An airplane will have wings, a tail section, a fuselage, etc. Each part may itself be composed of several parts as well--the tail will have a rudder and stabilizer, the fuselage will have a cockpit and engine compartment. At each level of detail, decide whether an item will make any difference in the finished image. You will eventually learn where to draw the line.

As you construct the parts, think of each one in terms of how it resembles `EDIT ADD` objects, or objects you have built before. Objects will often have several parts that are identical; never create the same part twice if you don't have to, that's what `DUPLICATE` and `REFLECT` are for. By thinking of a scene in terms of its parts, you can dispense with the 'easy' parts quickly, and concentrate on the parts that aren't so easy.

MORE TOOLS

Most of the time, you'll find that the 'hard' parts aren't that hard after all. The primary reason that a new shape looks hard is that you haven't done anything like it before. The best defense against this is to thoroughly familiarize yourself with the tools and commands that we have discussed, and the kind of shapes they can create.

Remember that once you have achieved a basic shape, there are still any number of ways to transform it into the object you want, by adding and deleting vertices, resizing, spinning, etc. Every new object you create will open up a whole new range of possibilities.

CHAPTER 7

THE WORLD

This chapter discusses the world in which Sculpt Animate objects reside. Facilities are described for lighting a scene and providing a ground and sky.

Creating lamps

The shadows, highlights, and subtle nuances of shading that give Sculpt Animate images a sense of reality depend upon the illumination of the scene. In chapter 2, we introduced the EDIT ADD LAMP command, which placed a lamp at the current cursor position. The lamp is depicted in the Tri-View as a white star.

There is virtually no limit to the number of lamps that can be created, but since each additional lamp increases the amount of computation required to generate an image, it is wise to try to use as few as possible. It's a happy coincidence that scenes with fewer lamps tend to look better, too.

For dramatic effects, you can employ colored lighting. In order to create a colored lamp, you must first use the EDIT MODIFY LAMPS command. This command calls up a requester similar to the one you used to modify the color of faces. Set the sliders to pick the color that you desire. There is no need to dim the light with the value slider. Instead you can enter a number into the lamp brightness box.

The brightness value given to a lamp sets its effect relative to other lamps in the same scene. If you like, you can think of this number as the wattage of the lamp, although the lamp's 'absolute' brightness is irrelevant. As we have seen, Sculpt Animate employs an automatic exposure technique in creating images, and its range of sensitivity is extremely large. For instance, a single ten-'Watt' lamp in a given scene should yield the same image as a single thousand-'Watt' lamp. If both lamps were in the same

THE WORLD

scene, however, the ten-Watt lamp wouldn't have much effect. In either case, you shouldn't have to worry about the overall brightness of the image.

If you have used the MODIFY LAMPS requester to change the current lamp color or brightness, then any new lamps that are created with the EDIT ADD LAMP command will have the new properties. If you indicate a lamp with the cursor before calling up the requester, you can use the two SET buttons to change either the color or brightness of the indicated lamp. If you wish to reset the properties of all the lamps in the scene, click on the button in the box titled WHICH LAMP, and the button will toggle between INDICATED and ALL.

You can use the two sets of FETCH and BLEND buttons, as appropriate, to fetch or blend lamp colors and brightness, much as you did with the colors of faces.

In Sculpt Animate, as in the real world, the brightness of a lamp falls off according to the inverse square law. This means that if you quadruple your distance from the light, the intensity falls to one sixteenth of its original value. Sculpt Animate's automatic exposure tries to find the best exposure for the most brightly lit object, so if you place a lamp close to an object, more distant parts of the scene may be very dimly lit. Try to avoid placing lamps much closer to one object than to others, unless you want to create some special effect. For an even, overall light, like sunlight, place your lamp five or ten times farther from the target than the observer's location.

If you want to move a lamp, you may do so with the Grabber tool described in chapter 4. First, be sure that all vertices in the scene are deselected, then indicate the desired lamp and invoke the Grabber as usual.

Try to resist the temptation to add lamps if a scene seems too dark. The automatic exposure feature will compensate for additional lamps, so they will usually not make a scene brighter. Instead, make sure the scene is lit evenly, with no objects too near a lamp. If you still want the scene lighter,

use the OBSERVER EXPOSURE MANUAL setting to brighten it.

To see the effect of various lighting arrangements, try rendering a few SKETCH mode images of a test subject. A one-subdivision white sphere does nicely. Start with a single lamp at or near the observer's location, then render a few images, moving the lamp around towards one side a little each time, until the subject is entirely side-lit. You will see that the shading of the sphere becomes deeper and more dramatic as the lighting becomes more oblique. It is usually good also to raise the main light up above the level of the subject, raising its highlights.

As the obliqueness of the main light source increases, so does the need for a second light--otherwise the shading becomes too deep. The second lamp, called a 'fill' light, will usually be placed to the other side of the observer, perhaps a little closer to the line of sight. The fill light needs to be dimmer. While it is all right to reduce its 'wattage' with the brightness setting, it may be easier to simply place it a little farther away from the subject, so you can see at a glance which light has more effect.

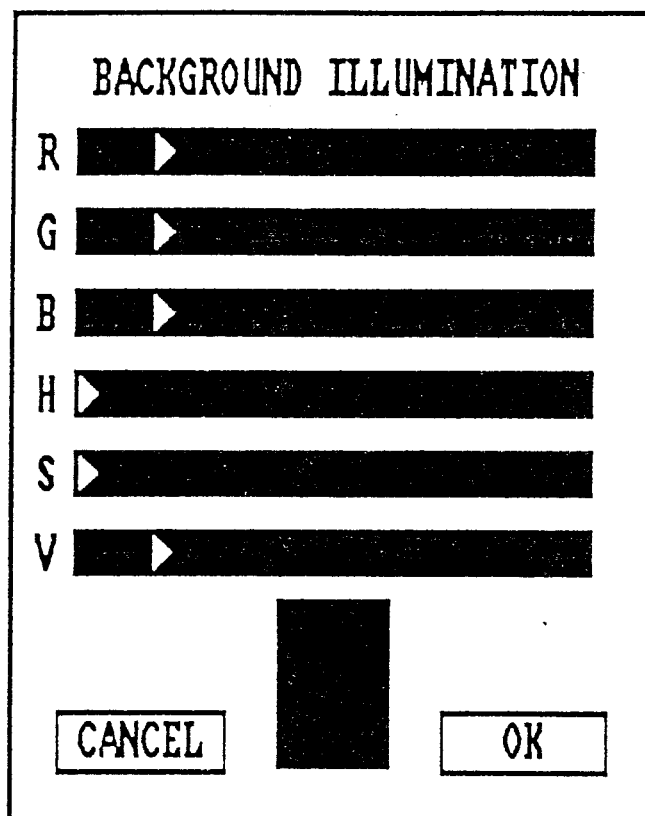
When you are using PHOTO mode, you will often want dramatic shadows to be cast on the ground (the 'ground' is explained later in this chapter). If the subject is on the ground, this can often be achieved by placing a third light source relatively high, and somewhat behind the subject. This 'backlight' should usually be as bright as, or brighter than, the main light. If the subject is above the ground or the main light is far enough to the side, then the shadows cast by the main and fill lights may be enough. In some cases the 'main' light may be discarded, and only a backlight and fill used.

These are not rules, by the way, just basic suggestions to get you started. The only rules when it comes to lighting are use your imagination, try things out, and remember the arrangements that please you.

THE WORLD

Illumination

Sculpt Animate employs a technique called low albedo approximation when it generates an image. This means that any part of an object that is hidden from all light sources will be totally dark. Such objects will not pick up reflected light from nearby illuminated objects. In order to soften the shadows, Sculpt Animate allows you to specify a diffuse ambient illumination that is independent of the lamps that you create.



The Illumination Requester

You can change the amount of the illumination with the command **WORLD ILLUMINATION**. This brings up an illumination color requester that you operate in the same way as other color requesters. Unlike the lamps, there is no separate brightness facility for controlling the intensity of the ambient illumination, so you must do this entirely with the value (V) slider. Usually the V slider should be set about one quarter of the way from the left.

The illumination setting can sometimes be used in place of fill lighting, but If the illumination value set too high, images will seem flat and lifeless. Too small a value will make shaded faces look unnaturally dark. The default value that is set at program start-up is a good all-around value.

The sky

You will have noticed a blue background in the images you have created so far. This background, although it is called the sky, need not be blue. Execute the menu command `WORLD SKY` and you will see a color requester pop up. The `SKY` requester is similar to the `ILLUMINATION` requester, but it has a couple of extra gadgets.

Unless it has been changed, the topmost gadget will contain the word `GRADUATED`. This gadget is a three-way toggle. Click on it a few times and watch it change from `GRADUATED` to `NONE`, then to `SOLID`, repeating the cycle as you continue to click.

If you leave this gadget set to `NONE`, there will be no sky, and the background will be black. Notice that the color swatch shows black, despite the settings of the sliders. Now select `SOLID`, and see that the swatch now reflects the color set by the sliders. If the sliders are moved, the swatch changes color normally. If you click `OK` and make an image, then the background color will be the one that you have selected.

Now select `GRADUATED`, and notice that an extra border is added around part of the swatch's perimeter. If the border surrounds the top half of the swatch, the sliders only change the color in that half. Now click anywhere on the swatch. The border should jump from the top of the swatch to the bottom, then vice versa if you click again. This is a special kind of toggle gadget. By selecting which half of the swatch is highlighted by the border, you can set up two entirely different colors. The upper color represents the sky.

THE WORLD

color at the zenith, or directly overhead. The lower color represents the color at the horizon.

If you make a picture with a GRADUATED sky and use SNAPSHOT or PHOTO mode, the sky color will blend smoothly between the ones that you have selected. In PAINTING and SKETCH mode, the color of the sky will be the zenith color.

As we have said, a GRADUATED sky blends the two selected colors into each other. In fact, some blending might actually occur if you were to set the two colors to the same value. This is because the color sliders are more accurate than the color resolution of the Amiga. If the slider setting falls between two usable values, Sculpt Animate 'dithers' the colors to simulate the correct shade.

Because a GRADUATED sky must be dithered anyway, this should cause no problem. If you select a SOLID sky, however, it will be assumed that you want no dithering, and the closest available 'pure' color will be used for the whole sky.

The ground

If you wish to depict a scene in which objects are not floating freely in space, but are sitting on or suspended near the ground, then you must make a model of the ground.

If the ground is to be flat, then a single triangular face could be employed. If it is a few hundred times bigger than the scale of the other objects, it will seem to stretch off to infinity. This sort of ground has some disadvantages: for one thing, there is the bother of constructing and placing the ground; for another, this type of ground will often obscure your objects in the SKETCH rendering mode (see imaging modes, chapter 3).

As a simpler alternative, you can use the ground plane that is built into Sculpt Animate. To use it, execute the command WORLD GROUND. A color requester will be

invoked. Like the sky color requester, it has a ground type gadget near the top right. By clicking on it, it can be toggled between NONE, SOLID and CHECKERED.

If NONE is selected, then no ground is produced. Otherwise an infinite ground plane is generated that passes through the Origin. The Origin is the center of the Sculpt Animate universe, and it is where the observer target is initially placed. If ground exists and it passes through the Tri-View, it is displayed as a horizontal line in two of the windows. If you enable the ground but can't see it in a Tri-View window, it is because the Tri-View 'box' is too far above or below ground level. Try expanding the Tri-View until the ground comes into view.

GROUND TYPE: **CHECKERED** 

R 

G 

B 

H 

S 

V 



CANCEL **OK**

The Ground Requester

If you select a SOLID ground, then it will be uniformly colored and you can use the sliders to pick a color. If you select CHECKERED, then the swatch divides into two and you can select two colors, just as you did for the sky. A checkered ground plane is one where alternate squares on the ground are different colors, like the squares on a chess

THE WORLD

board. The ground will only be checkered in the HAM imaging modes. In the other solid modeling modes, the ground will be the color of the top-half of the color swatch.

The size of the squares can be specified by clicking on the small checkered icon just to the right of the word CHECKERED. This sets the size of each square to be the same as the maximum dimension of the current Tri-View 'box'. For instance, if you click this gadget when the Tri-View is sized so that only a small portion of your object is visible, then the checkers will be small in comparison to your object. If the object looks small in the Tri-View when you click the gadget, then the checkers will be very large relative to your object.

In case you need to know, the center four-square of the checker pattern will always be centered on the Origin. The top half of the color swatch will be used for the northwest and southeast squares of this center block, the bottom half for the northeast and southwest squares.

Although a checkered ground plane has become something of a cliché in computer generated images, it does give the image more depth, and enhances the illusion that the ground extends to infinity. In addition, it adds interest to scenes that contain mirrored objects, for the reflections of the checkered ground make it easier for the eye and brain to perceive the shape of a mirrored object.

You can stretch the usefulness of a checkered ground with a little creativity. A red-and white checkerboard with fairly small squares can look like a tablecloth, especially if the observer is looking downwards. Enormous squares done in two earth-tone colors can suggest an agrarian plain. Additional features, such as hills, etc., can be added as objects placed in the background.

When you use a checkerboard, it is often best to keep the lamps fairly close to the subject, not much farther than the observer. This tends to darken the ground in the distance, reducing odd effects that can occur where the checker squares become very small. On the other hand, sometimes it

can add balance and interest to an image to place an extra lamp off in the background, illuminating a patch of the checkerboard where there would otherwise be an unbalanced empty space.

NOTES

CHAPTER 8

SPLINES

Earlier we showed you how to create basic curves in Sculpt Animate by placing individual vertices. This chapter describes a way by which you can take such a curve, roughly sketched, and easily transform it into smooth, sweeping curves. With a few mouse clicks you will be able to construct convincing profiles for aircraft and boats. Curves generated this way can be modified and controlled interactively.

Splines and knots

Splines provide a way of creating and manipulating smooth curves. In the days before computers, when a designer wished to generate a smooth curve, he used a thin strip of wood (or metal or plastic) and bent it to produce the desired shape. The strip was called a spline. Usually, the designer would mark a few points, called knots, on his paper, and bend the spline so that it passed through the knots. The spline defines the shape of the curve between the knots.

A mathematical simulation of a spline requires a little calculus, but all the messy details can be left to the computer. Sculpt Animate only needs three commands to work with splines: one command to make splines, one to modify them, and one to erase them. Although splines are very simple to use, there are some simple rules that must be followed, and since they are not at all obvious, you should study the next section quite carefully.

As we have seen, a curve may be represented by a sequence of vertices that are joined by edges. They are not really curved at all, they are just a sequence of straight line segments. If you use many segments, the approximation to a true curve becomes quite good. The curve tool can be

SPLINES

used to construct a sequence of edges, or the sequence can be built in a number of other ways using facilities we have already seen.

Having constructed a curve, you can make it a spline, but first you must tell Sculpt Animate which vertices are to become knots. To do this, be sure that all other vertices in the scene are deselected, then select a few vertices along the curve you are working with. The vertices you select will usually be key locations along the curve--points that you either don't want to move, or that you want adjacent vertices to follow.

Make sure that you have several unselected vertices between adjacent selected vertices. When the selected vertices become knots, the unselected ones will 'snap' into a smooth curve, and afterwards follow the knots wherever they go.

Now execute the EDIT DO MAKE SPLINE command. If you have done everything correctly, the command will subtly alter the curve you have laid out. If a warning message appears, you have probably selected vertices that do not all lie on a single, unbroken curve. If this happens no other action will be taken, so you can try again.

If the spline command succeeds, the unselected vertices on the spline curve will be re-aligned so they lie on a smooth curve which passes through the knots, which should not move. Each knot will have a small arrow drawn on the edges that connect to it. This identifies the vertex as a knot, even if it is not selected.

By making the curve a spline, you have given it a property of smoothness. Until the spline is erased, anything that you do with this curve will result in a smooth, flowing transition from one knot to the next. There is more than one way to draw a curve through a set of points, so the spline command picks one that follows a simple mathematical formula. You can exert quite a lot of influence over the curve, though, if you do not like the one that was chosen for you.

Let's try manipulating the spline some. Deselect all but one knot and use the grabber gadget to move that knot. If you watch closely, you will notice that the other knots do not move, but the non-knot vertices between the moving knot and the adjacent knots move so as to always lie on a smooth curve. Deselect the knot and select a non-knot vertex. Now try to move the vertex with the grabber tool and you will see that the grabber has no effect on a non-knot vertex of a spline. This is because the position of a non-knot vertex is solely determined by the positions of neighboring knots.

This is one of the exceptions to the rule that only selected vertices may be manipulated in Sculpt Animate. In this case, non-knot vertices, selected or not, may only be manipulated by working with adjacent knots.

Moving knots is the most fundamental way to manipulate a spline. Note that the spline does not have to lie in one plane. To see this, go to another Tri-View window and move a knot. You will see the spline twist into a third dimension. In Sculpt Animate, a spline is more like a piece of bent wire than a curve drawn on a flat surface. By moving the knots, the spline can be bent into almost any shape.

There is one tricky part to manipulating a spline this way. If the spline is very smooth (that is, if it has a lot of non-knot vertices), then it may be hard to select a knot by itself without contracting the Tri-View to 'zoom in' on it. Fortunately, this isn't necessary. As long as you've selected the one knot you want, selected non-knot vertices will just follow along as usual. This means you can use the Selector tool to 'shotgun select' several vertices in the neighborhood of the knot you want.

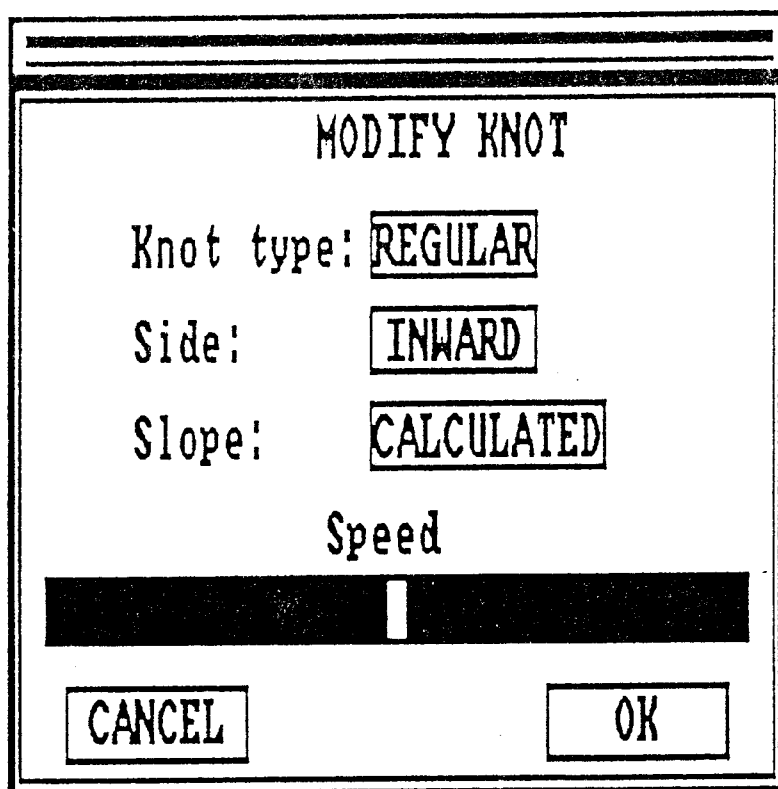
Slope and speed

The exact shape of the spline between two knots is determined by four parameters, two for each knot. One pair of parameters is called the slope and the other pair is called the speed. The slope at the knot is just the direction of the

SPLINES

curve in the vicinity of the knot, in mathematics, this direction is called the tangent. Unless you take special action, Sculpt Animate calculates a reasonable slope based on the positions of the knots on either side.

To see how this works, do an ERASE ALL to clear the Tri-View, and draw a curve with a dozen or so vertices. Deselect them all and then select the two end vertices. Make a spline, as described above. All the non-knot vertices will align themselves uniformly between the knots. With only two knots, there is no information about the shape of the curve that you would like, so Sculpt Animate generates the simplest possible curve between the knots--a straight line.



The Modify Knot Requester

Now move the cursor to one of the knots and execute the menu command EDIT MODIFY INDICATED KNOT. A requester with the title MODIFY KNOT will pop up. The requester has a drag bar, so you can move it out of the way if it hides the spline you are working on. In the requester, you should see a gadget marked CALCULATED, beside the title 'Slope:'. Click on this gadget, and it will toggle

between CALCULATED and SPECIFIED. Leave the gadget in the SPECIFIED state, and notice that a yellow arrow has now been drawn at the position of the knot.

The rotate gadgets in each window now have a special purpose. Instead of rotating selected vertices, they rotate the yellow arrow, which represents the slope of the knot. Rotate the arrow and notice that the curve between the knots deforms so that its tangent always matches the slope of the knot.

By switching to another Tri-View window, and using the rotate gadgets, you can point the slope in any direction and the curve will follow. If you click the slope gadget, so that it is in the CALCULATED state, the curve will snap back to correspond to the slope calculated from the positions of the nearby knots. The slope that you constructed is still remembered for the time being, in case you wish to return the knot to its SPECIFIED state later in the session. Only SPECIFIED knot states are saved with the scene, though.

When we manipulate a knot's slope, it is as though we were twisting a draftsman's wooden spline at one of the knots. Real splines would not long survive such treatment, nor do real draftsmen have surplus hands to apply torque at each knot, so our mathematical spline has creative advantages over its real-world counterpart.

The other control that you have over a knot is its 'speed'. This is set by the slider control near the bottom of the MODIFY KNOT requester. To best understand how this parameter works, specify the slope of the knot you've been working with so it's at an angle of about 45 degrees to the line between the knots. Now just move the speed slider and see what happens. Each time you release the slider knob, the spline will be re-drawn.

With large values of speed, the spline will seem to shoot out a long way before being influenced by the other knot. Small values of speed will cause the the spline to quickly forget about its initial slope and move towards the other knot. Also notice that the non-knot vertices get farther apart as the speed increases, and closer together as it decreases.

SPLINES

When you have finished manipulating the knot, you can either click on the OK gadget to accept its new settings, or click the CANCEL gadget to undo your manipulations. In either case, the requester will vanish, and the rotate gadgets will resume their normal operation.

If you make a spline with several knots and set the speed parameter for each knot to zero (the leftmost position for each slider). Then the non-knot vertices will move to lie on straight line segments between the knots. Try it. Although splines are usually thought of as smooth curves, they can be made quite angular.

Now go back and set the speed for each knot to its maximum value, and you will see that the spline shoots away from each knot with such force that it may even perform a loop before returning to the next knot. Quite baroque curves can be constructed in this way. Remember what they look like, you never know when they might be needed.

We can compare this operation to the use of a wooden spline. Increasing the speed at a knot is like making the spline stiffer at that point, while reducing the speed softens the spline locally. Our simulated spline has far outstripped the abilities of any real spline.

Although it is tempting to scatter many knots throughout a curve, the power of Sculpt Animate's splines is best realized if you use very few knots. If you go back to the example spline with only two knots, you will find that many very graceful curves can be created simply by adjusting their speeds and slopes.

In chapter 10 we will see how any 3D curve, including spline curves, may be used as a path to describe motion. In this use, each vertex of the curve represents a location in time and space. The 'speed' of an object following the curve will be directly proportional to the distance between vertices.

You should remember that this interpretation of the word 'speed' differs slightly from the 'speed' of a knot. While it is true that increasing a knot's speed generally moves the

adjacent vertices apart, the effect is local. The distance between vertices also depends on the speed of adjacent knots, and on the number of non-knot vertices between knots. Each of these factors will figure into the perceived velocity of the animated object.

Cusps

In mathematics, a cusp is a point where two curves meet with equal and opposite slopes, as in the horns of a crescent moon. In general usage, a cusp can refer to anything pointy looking. In Sculpt Animate, we will use a looser definition and apply the word to a place where two curves meet, but do not share a tangent.

To see how this applies to splines, create a spline with at least three knots, set the cursor on one of the interior knots, then call up the MODIFY KNOT requester. Look at the gadget labeled 'Knot type:'. As you may have noticed on other knots we have worked with, this knot's type is REGULAR. The 'type' gadget is a toggle gadget which toggles between REGULAR and CUSP--set it to CUSP.

When the knot is made a cusp, the gadget labeled 'Side:' will become effective. This is another toggle gadget that has two states, INWARD and OUTWARD. These states refer to the two arrows that mark the knot. One points inward, towards the knot, and the other outward, away from the knot. By toggling the Side gadget you can pick the side you want to manipulate. The side of the slope arrow that you can manipulate will be shown in yellow, the side that is shown in purple will not move.

When a knot is made into a cusp, it is split in two, so to speak. Each side of the knot affects the non-knot vertices on that side, but has no effect on the other side. This means that the knot suddenly has two slopes and two speeds, one set for each side. A cusp is like a point where the draftsman's wooden spline has been snapped and joined by a hinge. Use a cusp when you need to break an otherwise smooth curve with an angle or corner.

SPLINES

Using splines

By now you have probably thought of several uses for splines. If you need to build an object with smooth curves, a spline will ensure that the vertices are placed along a convincingly smooth shape. If your hand wobbles while you use the Curve Tool, the wobbles will find their way into objects that you Spin or Extrude from the curve. Picking a few strategically placed knots and creating a spline will even out all the wobbles.

If you want to modify a curve, it's much easier to grab a knot or two and move them, rather than have to individually move a dozen or more vertices and try make them represent a smooth curve. Of course, the trade-off is that non-knot vertices of a spline can only be moved by moving the knots.

If all the knots of a spline are selected, then the grabber will move the spline all at once, without changing its shape. The rotate gadgets may also be used freely on splines, and on sections of splines.

If you ever need to locate the knots of a spline, point the cursor at any vertex on the spline, and execute the command EDIT SELECT INDICATED SPLINE and each knot on the spline will be selected.

After you have built the curve that you need, it is a good idea to erase the underlying spline. Just move the cursor to one of the vertices of a spline and execute the command EDIT ERASE INDICATED SPLINE. The spline will be removed, but its vertices will remain. The vertices can now be manipulated in any of the usual ways. A spline uses up a certain amount of memory and it also slows down the refresh speed of the Tri-View, so once you are through manipulating the spline, it should be erased.

Splines can also be made that are closed loops. Create them in the usual way. For example, draw a square with the curve tool. Then use the EDIT DO SUBDIVIDE menu command a few times so that you have several vertices along each side. Deselect all the vertices; and select only the corner vertices as knots. Then execute the EDIT DO MAKE

SPLINE command and you should have a fairly tolerable approximation to a circle. Experiment with the speed parameters, and you will find that you can make the circle slowly revert to a square again.

An example: a gearwheel

Suppose you wished to make a convincing model of a gearwheel, you could use splines to make the job much simpler. First clear the Tri-View, and make the Tri-View windows large with the command EDIT DO MAKE TRI-VIEW BIG.

We are going to make a twelve tooth gear wheel, using eight vertices per tooth. In the west window add a circle with 96 subdivisions. Carefully select every fourth vertex of the circle. As a check, make sure there are three unselected vertices between every two selected ones, especially between the last and the first vertices that you selected.

Execute the command EDIT DO MAKE SPLINE. You now have a 96 vertex circular spline, with 24 knots (all the knots are selected). Now deselect every other knot, so only twelve vertices remain selected.

Execute the command EDIT DO EXPAND and make sure that the center of expansion is either CENTER or CENTROID. Move the upper slider a small distance to the left and watch a gearwheel shape emerge. When you are satisfied with the shape, leave the requester by clicking OK.

Move the cursor to any vertex and execute the command EDIT ERASE INDICATED SPLINE, because we no longer need to use the properties of the spline. You now have the profile outline of a gearwheel. Save this outline to disk--you will need an unfilled copy later.

Now the gear wheel profile may be filled with triangles, to make it a visible surface. If you were to select the gearwheel's profile and use the EDIT DO FILL command, the time to fill in a 96 vertex curve would be about half an

SPLINES

hour, so it might be quicker to use the edge maker tool. Alternatively, and perhaps less tedious, you can make a number of edges manually, to divide the outline into a dozen or so regions, then select each region in turn and use the FILL command on it.

In any case, once you have filled the profile, save the filled version to a separate disk file, being sure not to overwrite the unfilled copy.

You could now simply extrude the filled outline to create a solid gearwheel, but this step has two disadvantages. First, this will leave the back side of the gearwheel open. This is actually an advantage if it will never be seen, since it saves memory and speeds calculation, but it could be annoying if the back side needs to be there.

The second disadvantage only arises if you intend to use smoothing. There are two things that can confuse the Phong smoothing algorithm: flat surfaces made up of many faces, and right-angle corners (see chapter 4). If you envision our extruded gearwheel, you'll see it has both of these problems! If you wish to use smoothing on the curved portions of the teeth, you must somehow separate the gearwheel's edge from its flat surfaces.

One way to do this is to form the gear wheel from three unconnected parts: the two flat faces and the curved portion that joins them along the edge. This allows you to turn smoothing on only for the curved portion.

To do this, first clear the Tri-View and load the gearwheel profile that you saved earlier. Use MODIFY FACES to turn on smoothing, then extrude the profile. Add a copy of the gearwheel's filled surface with the PROJECT LOAD OBJECT command. Use the Grabber to move the face into place at the front of the gear wheel. Load a second copy of the filled outline and position it as the back surface.

Select the flat surfaces and use MODIFY FACES to turn their smoothing OFF. The gearwheel is complete!

CHAPTER 9

WHEELS WITHIN WHEELS

Whenever you must deal with a complicated system, it is always easier to break down the larger problem into a number of smaller, simpler ones. As we saw in chapter 6, it can be very convenient to think of a complex object as a collection of simpler parts. When the object is completed, you can connect the parts together and forget about them, dealing with the object as a single item again.

But what if we want to deal with an object as a whole at some times, and as a collection of parts at others? As we begin to animate objects in subsequent chapters, you will see ample need for this ability.

Sculpt Animate allows you to organize the objects of a scene, the parts of the objects, the parts of the parts, and so forth, into a hierarchy. At any given time you can deal with the level in the hierarchy that suits your needs. In later chapters you will see that, as with objects, object motions may also be placed in the hierarchy.

Hierarchies of objects

A hierarchy is a structure in which each member has a well defined level. Except for the uppermost level of the hierarchy, each member is subordinate to one superior member. On the other hand, each member may directly control one or more members at the next lower level.

This ordering is similar to that of a branch of the armed services, or a large corporation. The exception is that in these 'real world' hierarchies, lower members are often subordinate to any superior, not just one.

HIERARCHIES

Another everyday structure that takes the form of a hierarchy is a family tree. We will borrow some familiar terms like 'parentage' and 'offspring' when we describe the hierarchical structure used in Sculpt Animate.

The object structure of Sculpt Animate is imposed by giving names to groups of vertices. The names form a hierarchy. For the time being we will be discussing this hierarchical structure 'on paper.' The next section will discuss the mechanics of constructing and using such a structure.

Suppose that we have constructed models of several people, Fred, Joe and Bill, for example. It would be convenient to be able to treat the model of each person as a single entity, even though each may be constructed from several unconnected elements. It is as though we could issue the command "SELECT Bill", and have all the vertices of the model of Bill be selected. Bill could now be grabbed and moved to a new location, or rotated or expanded.

For many cases, this is quite enough. If, for instance, Bill were not a human figure, but actually one of three red regions on the hull of a flying saucer, then having a name tagged onto Bill's vertices permits you to select them independently, without having to care what they're connected to. This sort of ad hoc, piecemeal, naming of things can be very helpful if you are just modeling objects. More complex organization may not be needed.

But, as its name suggests, a big part of Sculpt Animate is animation. Bill is, more than likely, a human figure; or at least something that moves. If simple naming were all there was to the scene's hierarchy, Bill would be like a mannequin with no moving parts. In the hierarchy, though, we can arrange his parts into a series of levels, with the whole lot being named 'Bill'. The idea is to be able to select some of Bill's parts independently, or select Bill all at once.

Lets say we want to be able to select just Bill's left arm. It wouldn't be enough just to rename those vertices as 'leftarm,' because they would no longer be selected when

we select Bill. We need to have a way to name the vertices 'Bill's left arm.'

In Sculpt Animate parlance, this can be done by naming the vertices 'Bill+leftarm.' Assuming that we had deselected all vertices, the command "SELECT Bill+leftarm" would then cause all the vertices of Bill's left arm to be selected. By placing the cursor at Bill's left shoulder, the rotate gadgets could be used to raise Bill's hand, the first step in an animated salute, for example.

This suggests that we can decompose Bill into several elements, such as

Bill+leftarm
 Bill+rightarm
 Bill+head
 Bill+torso
 Bill+leftleg
 Bill+rightleg

Each element could be further decomposed, for example

Bill+leftarm+upper
 Bill+leftarm+lower
 Bill+leftarm+hand

The only limit to our subdivision is the amount of detail in the model of Bill. We can continue

Bill+leftarm+hand+finger1
 Bill+leftarm+hand+finger2
 Bill+leftarm+hand+finger3
 Bill+leftarm+hand+finger4
 Bill+leftarm+hand+thumb

If we compare the structure of Bill to a family tree, we can regard the name 'Bill' as the founding father with offspring 'head', 'torso', 'leftarm', etc. The name 'leftarm'

HIERARCHIES

begets 'upper', 'lower' and 'hand', and 'hand' produces five offspring in the fourth generation, including 'thumb'.

A drawback of this organization is that there may be combinations that we wish to select, like both thumbs, or all fingers and all toes, that are not accessible with a single command. When you plan your hierarchical structure, you should bear in mind what combinations are going to be used most often, and design the hierarchy accordingly.

Another major factor in the design of the hierarchy is motion. Although we have only touched on the subject so far, the primary use for a scene's hierarchy will probably be to animate objects. In later chapters, you will learn how motion fits into the hierarchy, but first you have to learn how to construct the hierarchy itself!

Building a hierarchy

As you have probably guessed, there is a facility in Sculpt Animate for building a hierarchy of names. To see it, execute the menu command `EDIT NAME SELECTED VERTICES`.

Although this command is designed to supply a name to a group of selected vertices, it also lets you build a hierarchy of names. The requester that pops up has a column of 9 slots. This represents the 'current generation' of the hierarchy.

Because we have not entered any names, all the slots are empty. To the left of the current generation is a long horizontal box titled 'Parentage', which is also empty. When the requester comes up, the box titled 'Add name:' will have a cursor in it, which means that it is ready for you to begin typing in the first name of the current generation. For starters, type in 'Bill'. When you press RETURN, the name will be transferred to the center slot of the current generation.

Now type in 'Fred', 'Joe', etc. As you type in more names, you will notice that they are entered into the current generation in alphabetical order, and the name is always entered into the center slot. You may wonder just what it is that you are naming. As of now, nothing. You are simply entering names into the hierarchy. You may enter a number of names all at once, then go back at your leisure and assign them to vertices.

The interface is titled "NAME SELECTED VERTICES". It contains a central list of names: "Bill", "Etc", "Fred", and "Joe". To the left of this list is a vertical scrollbar with up and down arrows. To the right of the list is a label "empty" with an arrow pointing to the top of the list, and a button labeled "OFFSPRING". Below the list is an "OK" button. On the left side of the window, there is an "Add name:" text box, followed by buttons for "CUT", "COPY", "PASTE", "Parentage", "RENAME", "ERASE NAME", and "CANCEL".

The Name Selected Vertices Requester

If you persist in adding names, a scroll bar will become active, so that you can move the list up and down, moving any name of the current generation into the center slot. If the name you want is among the nine items that are visible, you can quickly move it to the center slot by clicking on it with the left mouse button.

By now you will have realized that this center slot has special significance. If you were to terminate this requester by clicking in the OK gadget, any selected vertices would be given the name shown in the center slot. Call this name the current name.

HIERARCHIES

Now let's create a new generation. To start, click on the name Bill, or use the scroll bar to make Bill the current name. Next click on the gadget to the right, the one called OFFSPRING. Two things will happen. First, the column of names representing the current generation will be cleared. This is because we are looking at Bill's offspring, and he doesn't have any yet. The second thing to notice is that the name 'Bill' appears in the box titled 'Parentage:'. Bill is the parent of the yet to be created offspring.

To give Bill his 'offspring', add the names 'leftarm', 'rightarm', 'leftleg', 'rightleg', 'torso' and 'head'. Click on 'leftarm' and it will become the current name, or to be exact, the current name will be 'Bill+leftarm'. Click on the OFFSPRING gadget and we are ready to add the third generation: 'upper', 'lower' and 'hand'. The parentage box will show 'Bill+leftarm'. Continue in this way until you lose patience, or Bill's left arm structure is defined all the way to his thumb and fingers.

If you wish to backtrack to an earlier generation, click on the Parentage box. You are never able to add things to this box, but clicking on it will transfer you to the next older generation of names. If the list of ancestors becomes too large to fit in the Parentage box, you can still continue, but only the most recent ancestors will be displayed and three periods (...) will be displayed to show that the parentage list is not complete.

You have just constructed an elaborate hierarchy. As you create hierarchies of your own design, you should remember these important, if minor, restrictions:

- 1) The names that you use cannot be more than 10 characters long (that is the most that the 'Add name:' gadget has room for).
- 2) You may use upper or lower case letters, but the name will be treated as though it was all in upper case, for the purpose of maintaining the current generation list in alphabetical order.

HIERARCHIES

- 3) The same name may not be used more than once as the offspring of the same parent. Thus Bill can only have one 'leftarm' on any one level, but Bill and Joe can each have a 'leftarm' of their own.
- 4) If the name begins with a digit (0 through 9), strange things may happen. Avoid using names that start with a digit until you learn the ramifications of doing so (in chapter 12).
- 5) Do not use any characters except letters, digits, period (.), underscore (_), minus (-) or sharp (#).
- 6) The object hierarchy may be up to 16 generations deep, but you will probably never need more than 5 or 6 generations.
- 7) A name may have up to 255 offspring, which should suffice for the most prolific name.
- 8) Avoid using the names LOCATION and TARGET, these names are reserved so that you can refer to the observer's location and target for animation purposes.

If you break these rules, the name will usually not be added to the current generation of names. The exceptions are rules 4 and 8.

Although the same name cannot appear twice in the same level with the same parentage (rule 3), the same name can be used on different levels of a branch, for instance, Bill's upper and lower arm may both be named 'arm' in this fashion: 'Bill+leftside+arm+arm'. The names are on different levels, so they don't conflict. The name 'leftside' in this example might not even name any vertices by itself. Such a 'null' name can be used as an attachment point for grouping other named vertices.

HIERARCHIES

If you make a mistake entering names, the current name can be erased with the ERASE NAME gadget, provided that the name has no offspring. If it has, erase all the offspring first. Any vertices that had been named by the erased name become unnamed 'free agents'.

At any time you can leave the requester by clicking on the OK or CANCEL gadget. The OK gadget will assign the current name to any selected vertices. The CANCEL gadget allows you to leave the requester without assigning any name to vertices, but it will not destroy any changes you have made to the hierarchy.

It is possible to assign a name to vertices that have already been named. If this happens, the old name will no longer apply to the renamed vertices, since a particular vertex can have only one name. Suppose that the vertices of Bill's left thumb were selected, and you named them 'Joe', then no vertices would respond to the name 'Bill+leftarm+hand+thumb'.

In our example, we have entered several names into the hierarchy that do not, as yet, have vertices assigned to them. When you are building an object with a specific plan in mind, it works well to lay out the hierarchy beforehand and enter it all at once. The vertices of your objects may be selected and assigned names afterward. It is just as well in some cases to create your objects or parts of objects first, then assign names to the various parts. You will soon find which approach suits your style most in various situations.

If you have constructed your object so that its parts are not connected by edges, it may be easier not to select vertices for naming, but to simply indicate the object you wish to name with the cursor and use the command EDIT NAME CONNECTED VERTICES. This command is similar to EDIT NAME SELECTED VERTICES except that it operates on vertices that are connected to the indicated vertex.

Using an object hierarchy

The object hierarchy can be used to select, deselect and erase groups of vertices by name.

Going back to our example hierarchy, you may select Bill's left thumb by executing the menu command EDIT SELECT NAMED VERTICES. A requester very similar to the NAME SELECTED VERTICES will pop up. Instead of the 'Add name:' and ERASE NAME gadgets, there are SELECT and DESELECT gadgets.

To use this requester, you must first pick a name. Do this by clicking on a current generation, Parentage, or OFFSPRING gadget. When the current name is correct, press the OK button, and the selection will be made as you leave the requester.

You don't have to leave the requester every time you want to select something. Instead, you may press the SELECT gadget, which immediately selects the vertices named by the current name, as though you had clicked OK and executed the command again. This leaves you free to change the current name and select something else.

As a matter of convenience, there is also a DESELECT gadget which allows you to deselect vertices without leaving the requester. For example you could select all four of Bill's left fingers (and his palm, if he has one) by choosing the name Bill+leftarm+hand and clicking on SELECT. This also selects Bill's thumb, though, so next you can click OFFSPRING, then 'thumb' to make it the current name, and then click DESELECT.

The SELECT and DESELECT gadgets take effect immediately. If you press on the CANCEL gadget to leave the requester any selections or deselections that you have made with them will be retained. When you click OK to leave, vertices with the current name will also be selected.

If you try to select vertices that are already selected, the screen will flash to tell you that your request had no effect.

HIERARCHIES

The menu command EDIT DESELECT NAMED VERTICES calls up a requester that works the same as its counterpart, except that the OK gadget causes vertices with the current name to be deselected instead of selected. It also has both SELECT and DESELECT gadgets.

The command EDIT ERASE NAMED VERTICES invokes a requester called ERASE NAMED VERTICES. If you click OK in this requester, the vertices named by the current name, as well as those named by all its offspring, will be erased.

The ERASE NAMED VERTICES requester has two special purpose gadgets: ERASE VERTICES and ERASE NAME. ERASE VERTICES has the same effect as clicking OK and then re-invoking the requester. ERASE NAME erases the current name from the hierarchy, unless the name has offspring. To erase a name with offspring, erase all the offspring names first.

If you wish to erase a name and its vertices, be sure to erase the vertices first. Once the name is erased, you will have to select the vertices by some other means before you can erase them.

Editing the hierarchy

Mistakes can happen and it is important to be able to easily correct them. If you wish to alter a scene's hierarchy, call up any of the EDIT NAME requesters, such as EDIT NAME SELECTED VERTICES. In addition to their normal functions, these requesters permit you to re-order and otherwise edit the hierarchy.

For instance, if you want to change the name of a hierarchy element, select the name that you wish to change and press the RENAME button. A requester will pop up to let you type in a new name to replace the current name.

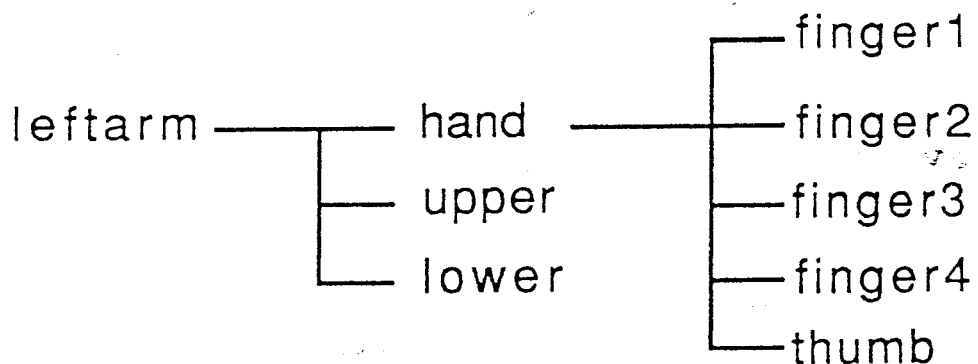
These requesters also have CUT, PASTE and COPY buttons. The command CUT will extract the current name

and its vertices from the hierarchy, along with all its offspring and all the vertices they name. These items are stored in a temporary location in RAM.

The COPY button works in a similar fashion, except that the items are only copied into temporary storage, and the original vertices and names are unchanged.

The PASTE gadget may be used to retrieve items that have been stored with COPY or CUT. When PASTE is clicked, the name that was current when the last CUT or COPY occurred will be entered into the current level of the hierarchy. All that name's vertices will also be entered, as well as its offspring and their vertices. Note that the copied vertices will lie in the same location as the original vertices, so you will probably want to select these vertices, grab them, and move them somewhere else.

The hierarchy action of CUT and PASTE should become clear if we use these commands to supply an object named 'Joe' with a left arm copied from 'Bill'. First issue an EDIT NAME SELECTED VERTICES menu command. Make the name Bill+leftarm the current name and click on COPY. The vertices named Bill+leftarm will be copied to temporary RAM storage, along with the matching hierarchy fragment, something like this:



Now move up a level and select or type in the name 'Joe', then click on the OFFSPRING gadget. Currently Joe does not have any offspring, so all the current generation slots should be clear. Now press the PASTE gadget. You will notice that Joe now has the hierarchy structure of a left

HIERARCHIES

arm. If Bill's left arm actually contained vertices, these have been duplicated.

When you perform a PASTE, it is possible that the item you want to paste will have the same name as another item in the same level of the hierarchy. If this happens, you will be asked if you want Sculpt Animate to merge the two items. If you say no, a new name will be generated for the new item, which you can rename later if you wish.

If an item is pasted farther down in the hierarchy than it originated, it is possible that it's structure might extend past the sixteen level limit. If this occurs the hierarchy segment will be truncated to fit. New vertices that would have been named by the truncated levels will be unnamed.

Loading and saving named objects

Two more commands serve to round out the object hierarchy system. The menu command PROJECT SAVE NAMED OBJECT, invokes a simplified version of the standard hierarchy requester. After you have selected a name from the hierarchy by traversing the family tree, you can press the OK gadget to save the currently named object as a scene file. A standard file requester will appear to let you enter or choose a name for the scene, just as if you had issued the PROJECT SAVE OBJECT command. The part of the hierarchy that corresponds to the object will also be saved in the scene file, just as it would be stored by CUT or COPY.

The complementary command, PROJECT LOAD NAMED OBJECT calls up a requester similar to the NAME SELECTED VERTICES requester. You can use the usual gadgets to modify and traverse the hierarchy. When you select a particular level in the hierarchy and click on the OK gadget, you will first be prompted for the name of a scene file, and then the object will be loaded. If the scene file contains hierarchy information, for example if it was produced by the PROJECT SAVE NAMED OBJECT command, then the hierarchy will be merged into the current

HIERARCHIES

hierarchy at the selected level. It doesn't matter which member of the current generation was chosen, the loaded object's hierarchy will just be loaded into the selected level.

CHAPTER 10

MOTION PATHS

The last chapter hinted that a scene's hierarchy may be used to give an object motion. This chapter explains how motion can be created and manipulated in the Tri-View as a special sort of object called a path. Paths are an instrumental part of global animations, one of Sculpt Animate's two fundamental types of animation, both of which are described in the next chapter.

Paths

We have seen that, in Sculpt Animate, a set of vertices linked by edges can be used to represent a curve in three dimensional space. We have also seen how splines are power tools for creating smooth curves.

Sculpt Animate also supports a slightly different entity called a Path. A path is used to represent the motion of a point that moves through space. If you regard time as a fourth dimension, then a path is a 4 dimensional curve. This may sound rather frightening. Fortunately, paths have a simple representation in Sculpt Animate, and no mathematical knowledge is required to use or manipulate them.

Instead of the abstract concept of a point moving through space, let us take a concrete example. Suppose we wish to make an animation of a man running across the Sahara desert. Perhaps through derangement caused by lack of water, he does not run in a straight line but in irregular loops and wobbles. What's more, he not only moves in the horizontal directions, but also up and down the sides of the desert's sand dunes. He is a living model of three dimensional motion.

MOTION PATHS

His footsteps in the sand leave evidence of where he has been. If we assume that he is running at a steady pace, say one pace per second, then by counting footsteps we know when he was at a particular place. Footprint number 60 must have been made one minute after he started running. His motion has defined a clear, unambiguous path.

In Sculpt Animate, we can mimic this sort of path by placing a vertex at each location, or 'footstep', and link adjacent steps with an edge. This will just be a curve in space, but we can interpret it as a path if we assume that each vertex fixes the location of our modeled man at a particular time. While the real man's hapless wanderings apparently created his path, our imaginary man will follow the path we create for him.

Once we have described the path for our man, or any moving object, Sculpt Animate will create a sequence of images, similar to the ones we have been creating all along. Each image shows the animated object in a slightly different place along the path. When the images are shown in rapid sequence (many times per second), the object seems to move along the path.

Each of these images is called a 'frame' in animator's parlance. Instead of dividing up our paths by footsteps, we say that each vertex on a path represents the animated object's location in a particular frame of the animation. For smooth motion, of course, there would probably be more than one vertex (i.e., frame) for each step along a running man's path.

Creating paths

We have already seen that it is easy to create a curve in Sculpt Animate. A curve is not quite the same thing as a path, however. The primary difference is that a path must have an unambiguous starting point. A closed curve path (like a circle) could start at any vertex, and go either direction around the curve. An open curve path could start at either end.

MOTION PATHS

Another important rule for paths is that they must never fork. The reason is clear enough, if the vertex defining one location of the path has two edges leading away from it, which way is the program going to move the object?

Because of these considerations, we need to tell Sculpt Animate that a curve is to be a path, where it starts, and--in the case of closed loops--which way it leads.

If the curve is open (has two ends), you can do this simply by moving the cursor to one of its end vertices and executing the command `EDIT DO MAKE PATH`. The cursor indicates the beginning of the path, where the object will be placed in the animation's first frame.

If the path is a closed loop or circle, then you need to pick a direction for the path, as well as indicating the first vertex. This is done by first executing a `DESELECT ALL`, then selecting the vertex that will be the *second* location in the path. When this is done you can indicate the first vertex and issue the command `EDIT DO MAKE PATH` as usual.

Sculpt Animate will examine the indicated curve to make sure it is suitable for use as a path. If it is not, the program will display a message telling you why, and nothing else will happen. If the curve is suitable, a small arrow will be drawn halfway between its first and second vertices, to show you that it has, indeed, been made a path.

Except for memory, there is no practical limit to the number of paths you may create. Later in this chapter, you will see how to tell Sculpt Animate which path a particular object is to follow. You will also see how the path an object follows may be attached to yet another path, so that very complex movements may be described.

If you ever wish to erase a path, you may do so by erasing its vertices. If you want to erase the path, but retain its vertices and edges, indicate any vertex on the path and issue the command `EDIT ERASE INDICATED PATH`.

MOTION PATHS

If you want to see if a curve is a path, but don't want to bother looking for its little arrow, indicate a vertex on the curve and call the command EDIT SELECT INDICATED PATH. If the curve is a path, its vertices will be selected. If not, a message will be displayed to tell you so.

Tumbling

The last section told you how to create a path to describe the motion of an animated object. The next section will tell you how to make the object follow the path. This section deals with an important detail which lies between these two problems: how will the object be *oriented* as it follows the path?

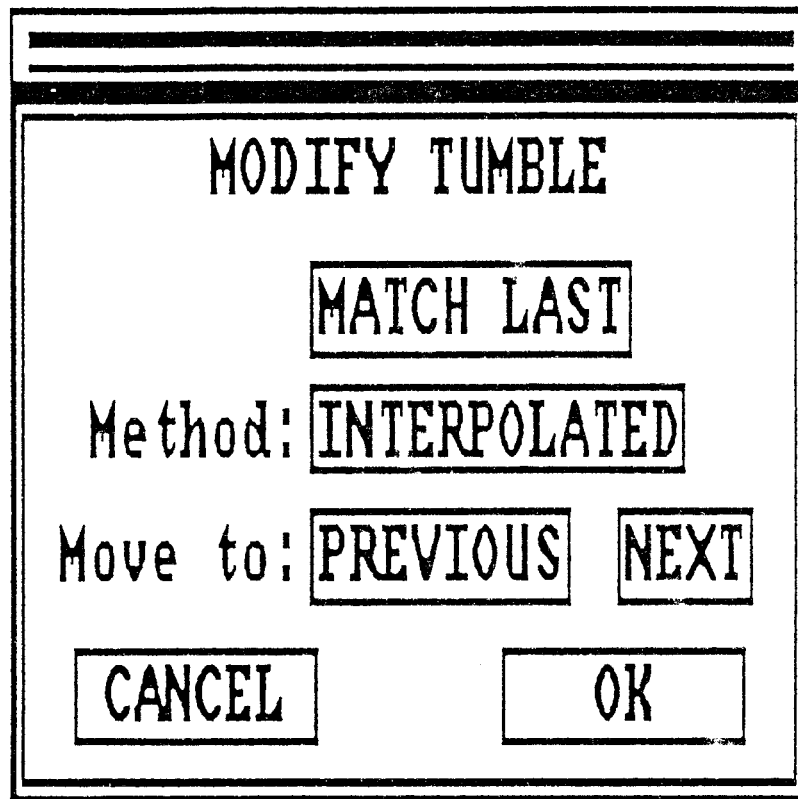
Let's go back to our example of the running man. Ignoring for the moment the rather difficult problem of leg and arm motion, let's look at how the man should position his body as he runs. For one thing, it is obvious that he will always face more or less in the direction he is running. Furthermore, he will probably pitch to one side or the other when he turns. Finally, most people tilt forward somewhat as they run--more or less so, perhaps, when climbing or descending a sand dune.

If you were to attach our imaginary man to a path created using just the techniques of the last section, this is not the sort of motion you would see in the resulting animation. Rather, the man would slide blindly along the path, always facing in the same direction he faced when he was constructed. This is because the program has no idea what a man is, or how he runs--if it did, it would be a much more complex program indeed!

Rather than trying to guess how the objects you create will move, Sculpt Animate simply allows you to specify an object orientation for each vertex along the path. This orientation is called the 'tumble' for that location. When an object follows a path, it orients itself according to the tumble setting for each vertex.

You can set the part of the object about which it will tumble (see the 'Local Origin' section, later in this chapter), so you have complete control over the object's motion.

If you specify the tumble for only a few vertices, Sculpt Animate can often guess the correct tumble for the ones in-between. We'll speak more of this later in this section.



The Modify Tumble Requester

To specify the tumble setting for a vertex of a path, first move the cursor to the vertex and execute the menu command EDIT MODIFY INDICATED TUMBLE. This command brings up a requester titled 'MODIFY INDICATED TUMBLE'. Like the MODIFY INDICATED KNOT requester, this one also has a drag bar on the top so you can move it around as needed.

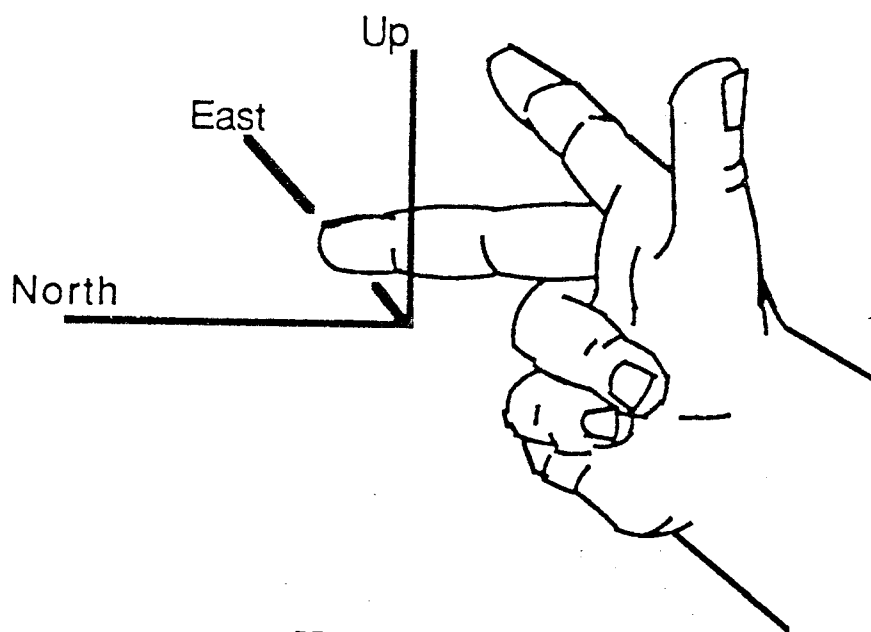
When the requester comes up, the indicated vertex will become centered in the Tri-View, and three brightly colored lines will be shown extending from it. These lines represent the Tumble Axes for that vertex of the path.

MOTION PATHS

The first time you call up the requester, one line will be pointing East, one will be pointing North and one will be pointing upwards--so that only two of the lines are visible in any one window. At this time, the lines all seem to have the same length on the screen.

Let us call these lines the East Tumble Axis, the North Tumble Axis and the Up Tumble Axis, respectively. Note that the East Tumble axis is yellow, the North Tumble Axis is White and the Up Tumble Axis is purple. You should remember which color corresponds to each direction.

As an aid to memory, you can paint your right index finger yellow, your middle finger white and your thumb purple. With a little contortion you can extend the two painted fingers and the thumb so that each is at right angles to the other. If you keep your thumb upright and point East with your index finger, your middle finger will be pointed North. Just like the Tumble Axes.



The Tumble Axes

Now twist your wrist and turn around. Naturally, your index finger (the yellow, East, tumble axis) can be pointed in any direction. You can tilt it up or down, or swing it left or right. Your index finger is said to have two degrees of freedom, so far as pointing is concerned.

If you decide to point the East tumble axis (yellow) in a particular direction--towards a lamp, for example--you will suddenly find that the thumb and middle finger (purple and white, the Up and North axes) can only be rotated around the East axis, so long as you hold them at right angles. These axes are reduced to one degree of freedom.

Once you decide on an angle for one of the two remaining axes, say North (white), then the last axis (Up, purple, your thumb) is also fixed, having no degree of freedom at all.

The paint on your hand should be dry enough now that we can go back to Sculpt Animate, where the Tumble Axes for one vertex of your path have been waiting patiently. We will assume that they are still in their original orientation.

While the MODIFY INDICATED TUMBLE requester is present, you can use the rotate gadgets in each Tri-View window to twist these tumble axes. For instance, click the counterclockwise rotate gadget in the North window a few times, then click the same gadget in the Down window several times. The yellow axis will be tilted up a bit, then rotated towards the North. The white axis will still be level.

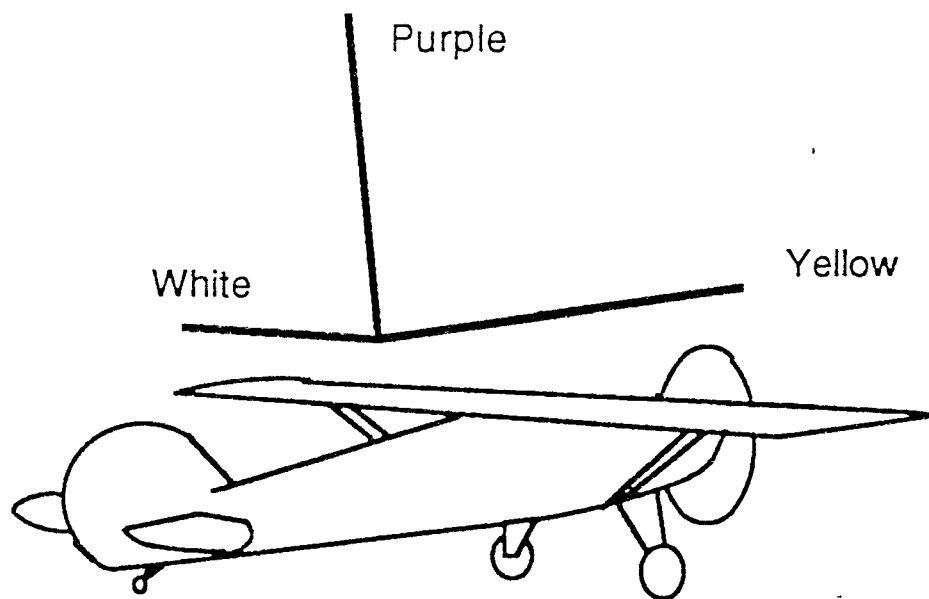
In addition to rotating, you will notice that each axis line has become visible in the one window where it couldn't be seen before. As they are rotated, the axes will grow and shrink in various windows according to their orientation. This provides additional visual feedback.

Yet another change will occur when you first rotate the Tumble Axes for a particular vertex. This change takes place in the MODIFY TUMBLE requester itself. The toggle gadget titled 'Method:', which was previously marked INTERPOLATED, will change to SPECIFIED. This means that the tumble you have set for this vertex will be used. Otherwise, Sculpt Animate will try to guess (interpolate) the correct tumble by picking an orientation between the two adjacent SPECIFIED tumbles. You may click the toggle gadget back to INTERPOLATED if you change your mind.

MOTION PATHS

To visualize the effect of the change we just made to the Tumble Axes, let us abandon our poor man in the desert and take up the case of a search plane trying to find him. Let's say the plane had been built facing East, with the wings level.

If this plane were to fly along the path we've been working with, then when it reached this location on the path its nose would be pointed up a bit and it would be pointed somewhat North of East, with the wings still level. In other words, it would be positioned just like the Tumble Axes for this vertex.



Tumble Axis Relationship

Of course, if you only set the Tumble Axes for this one vertex, then the plane will probably not fly very naturally. As we have seen, you don't need to set a tumble axis for every vertex on the path, just for certain important locations, but you almost certainly need to set more than one. For instance, you will probably want to specify the tumble for the first and last position on a path.

Let's imagine that you have created a new 'S' curve path for the airplane. (This can be done with a two-knot spline, try it!) Typically, a plane will enter this type of maneuver flying straight and level, then tilt its wings in the direction of the first turn (called 'banking'), level out again between

turns, bank the other way into the second turn, and finally level out again.

Likely places for specifying the tumble are the first and last vertices, the midpoint between curves, and the steepest portion of the bank in the middle of each curve.

The first, last, and middle tumbles are easy, since you only have to rotate around one axis. Just indicate each of these vertices and call MODIFY INDICATED TUMBLE, then rotate the axes for that vertex in the Down window until the yellow (East) axis is pointing along the path.

Setting the tumble in the middle of the curve is a little harder. It's a good example of a typical tumble axis problem. You want the yellow axis to point along the path here, too, but you want the other two axes to be tilted sideways, to lean the airplane into the curve. So far you've been working in the Down window, but now you need to move to one of the side views, North or West. Go to the side view where the yellow axis looks *shortest*, and rotate the axes in the desired direction. If the yellow axis looks the same in both side views (the worst case), just pick one at random.

If the yellow axis was visible at all in the view you used, you will notice something a little troublesome. The rotate gadgets in a Tri-View window can only rotate in the window's frame of reference. Because of this, if the yellow axis is visible, there's a good chance that it will tilt either up or down when you rotate the other axes. To correct this drift, go to the *other* side view and straighten the yellow axis. This may 'unwind' your initial tilt a bit, but by switching back and forth a couple of times you will get the right position. When you get the right amount of bank, you may also have to correct the heading in the Down window.

This is a general rule. If you want to rotate the Tumble Axes around one axis, use the window where that axis is least visible. If that axis drifts anyway, correct it in the other two windows. By repeating this cycle a couple of times, you can get the right orientation fairly easily.

MOTION PATHS

There's a trick that you can try until you get used to this. If you want to rotate to a certain position around one axis, align that axis with the Tri-View so that it disappears in one window. You can now set the rotation easily in that window. Once this is done, it is fairly simple to point the remaining axis in the correct direction. In the case of our airplane, for instance, you would tilt it up or down in one of the side views, then set its heading in the Down view. You will still get some interplay, but it shouldn't be bad unless the plane climbs or dives radically.

Once you have set the tumble axes for some representative locations, it is a good idea to check the vertices in-between. The NEXT and PREVIOUS gadgets on the MODIFY TUMBLE requester let you step forwards and backwards along the path, examining the tumble axes to see if they are being interpolated the way you expected. In this way you can 'test fly' the path. If the axes seem to stray, move to the vertex where they seem most out of line and set it as you wish, then examine surrounding vertices to make sure they fell into line, too.

A useful gadget that hasn't been mentioned yet is MATCH LAST. It will swing the Tumble Axes around to be aligned with the last set of axes that you positioned with the requester. If you are working your way along a path with the intent of rotating the axes a specific amount for each vertex, then you will find that the interpolation often anticipates you, messing up the count. To ensure a good count, click MATCH LAST before setting each vertex.

MATCH LAST can also be used to copy Tumble Axes from one vertex to another. This is good for matching the start and end of a path for looping animations, or for copying Tumble Axes from one path to another. First, call MODIFY INDICATED TUMBLE for the 'source' vertex, then click *both* rotate gadgets in any window, once each. This does nothing, but makes the program remember that location's tumble. Finally, go to the 'destination' vertex and click MATCH LAST.

For more precise work, note that the rotate gadgets may be used with the Amiga and ALT keys as usual. The normal rotation is 5 degrees per click. When clicked with the left Amiga or ALT key pressed, rotation is 1 and 0.1 degrees, respectively. The right Amiga and ALT keys make rotation 45 and 180 degrees per click, respectively.

Motion hierarchies

We have seen how to create a special 'path' object and set its Tumble Axes, to make another object 'attached' to the path appear in a specific location and posture in every frame of an animation. So far, we have neglected to mention how the object is attached to the path.

We have also seen that objects and parts of objects may be organized in a scene's hierarchy, so that objects may be 'attached' to one another with the lower level objects 'following' the higher level ones. We have hinted that motion can be placed in the hierarchy somehow.

By now, you've probably guessed how--just put a path in the hierarchy! Any object that is named as a path's offspring will follow that path.

There is a special command for naming paths: EDIT NAME INDICATED PATH. Before calling it, indicate a vertex on the path you want to name. Except for the title, the requester that pops up is identical to NAME SELECTED VERTICES, with gadgets for traversing the hierarchy and entering a new name. When you click on the OK gadget, the path will be named and placed in the hierarchy.

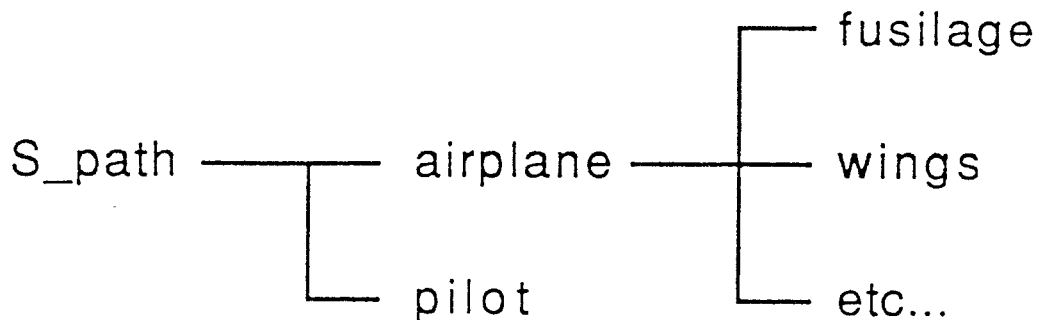
If a name in the hierarchy names a path, then when it is in the 'current' slot the label 'path' will be shown with an arrow pointing to the name.

A path differs slightly from other objects in that only the one path can be given a particular name on any given level. If any other vertices already have the exact same name, they will be 'unnamed' when the path is named.

MOTION PATHS

On the other hand, if you give some vertices a path's name, then the path will just be demoted to 'vertices' in the hierarchy.

Any number of named objects can be attached to a path, using EDIT NAME SELECTED VERTICES to name them as offspring of the path. All the offspring will move as a unit. Consider this example:

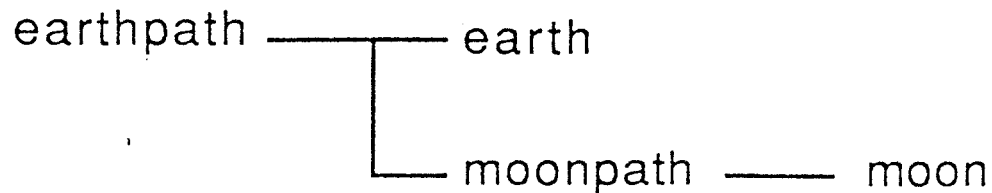


The airplane, which is composed of a fuselage, wings, etc., is attached to the object named 'S_path'. If this object is a path, then the airplane will follow it. The object named 'pilot' will also follow along the path, moving as though it were a part of the airplane.

This shows that you can group objects under a single name and attach that name to a path, or just attach all the parts directly to the path. It's just a matter of convenience. The former scheme is often easier, but either way there are some important considerations about where parts will appear relative to each other and to the path. The next section explores this.

We have been describing paths as a special sort of object. Since objects can be attached to paths, can't paths be attached to paths? Yes! The 'lower' path and its object will simply follow the 'upper' path, like a flea walking on a dog's back. This is called 'hierarchical motion', and it allows very complex motions to be described easily.

As an example, let's imagine a simplified orbital model of the Earth/Moon/Sun system:



The name 'earthpath' can be given to a 364 vertex circular path around the Sun. This path's name goes in the top level of the hierarchy. Two objects are attached to 'earthpath': one is the earth object itself, the other is 'moonpath', a 28 vertex circular path around the earth. The moon is attached to 'moonpath' (i.e., its offspring).

Notice that the number of days per year and month have been 'fudged'. This allows a smoothly looping 364 frame animation (hard disk recommended!). More accuracy would require much denser paths, and a 119802 frame loop!

You might wonder why the sun isn't in the hierarchy. It doesn't really have to be, since it doesn't move. Of course, it could be named in the top level for editing purposes, if you like.

A better question might be how to get the two paths to be centered around the Sun and Earth objects correctly, and how to get the planets to follow their paths accurately. This is not just a matter of placement in the Tri-View, as we'll see in the next section.

If you're curious, this model could be done fairly close to scale in Sculpt Animate, but it would be snug, and the scales are so large it would be hard to make a very exciting animation with it. Chapter 13 will show you how to use coordinates to scale things. Using two decimal places in the coordinates window, the Moon could have a radius of 1.00 units. The Earth's radius would then be 3.67, the Sun's, 400.36. Moonpath's radius would be 221.14, and earthpath's would be 86043.84. The model isn't perfectly accurate, but it far exceeds the resolution of the Amiga's screen.

MOTION PATHS

By the way, these dimensional scales were derived from the CRC Handbook of Chemistry and Physics, Robert C. Weast, ed. (CRC Press, 1975), pp. F-171+.

Local Origins

So far, we have taken it for granted that when an object is attached to a path, it moves 'along' the path, matching the location of each path vertex in turn. But what does that mean, exactly?

As our man runs in the desert, we might expect him to run 'on' his path, so the soles of his feet match each path vertex in turn. If the airplane is moving along the path, we might expect something different.

The point is, Sculpt Animate has no way to guess what we expect, so we have to specify a certain point, relative to any object, that will coincide with the 'active' vertex of the object's path in any given frame.

This point is called the object's local origin. It is analogous to the guide pin on an old-fashioned slot car. While the car might swing wildly about rounding curves, the guide pin always follows the slot precisely (until it falls out). The guide pin, like a local origin, serves a dual purpose. It holds the car 'object' to its 'path', and it acts as a pivot for the car's 'tumble'.

When an object is created, its local origin matches the center of the Sculpt universe, the 'absolute origin'. When you run Sculpt Animate, this point will be in the center of the Tri-View.

Because of this, if you simply run the program and begin building an object to animate, the object's local origin will probably wind up somewhere near the object's middle. This may do perfectly well for simple animations.

When you begin to animate multiple objects, or objects that need to tumble or pivot in a particular fashion, then you

will need to remember to set a local origin for each animated object.

To do this, first move the cursor to the place, relative to the object, where you want its local origin to be. Next, execute the menu command EDIT MODIFY LOCAL ORIGIN. A familiar looking hierarchy requester will pop up. In this requester, find the name of the object in question and move it into the current name slot. Finally, click on the SET gadget. The local origin will be set to coincide with the cursor position. If you decide not to set a local origin, click CANCEL.

If, for any reason, your object's motion is not what you expected, be sure to double-check its local origin. You can do this by making its name current in the MODIFY LOCAL ORIGIN requester and clicking SHOW.

A local origin that was placed with SET will stay where it is in the Tri-View, no matter how you move or modify the associated object's vertices. This lets you fine tune and edit an object even after its local origin is set.

If you actually move an object, however, you may want the object's local origin to move along with it. If you think an object will be moved around after you set its local origin (for instance, if you will be loading it into several different scenes), then you can lock the local origin to a particular vertex by indicating the vertex, then using the LOCK gadget instead of SET. Whenever that vertex is moved in the Tri-View, any local origin locked to it will follow.

Usually, the vertex you LOCK to will be one of the local origin's named vertices, but it doesn't have to be. This is especially true of paths used in hierarchical motion. If you were to LOCK a path's local origin to one of its own vertices, it might not follow the parent path as you wish, but you cannot give any other vertex the path's name. In this case, the path's local origin would probably be locked to a vertex of some object on the same level as the path.

MOTION PATHS

Remember that you only need to set the local origin for objects that move, and that the local origin is part of the object, not of the path it follows. A path only needs a local origin if it follows another path.

For instance, take the orbital example of the last section. 'Earthpath' is highest in the hierarchy, and it does not move in the universe. It does not need a local origin. 'Moonpath', as well as the 'Earth' and 'Moon' objects, all move, so they all need local origins.

In the orbital example, it would be simplest to select each of the objects in turn, call SNAP CURSOR TO CENTER, and SET the object's local origin.

Each of these objects is either circular or spherical, so it makes sense for their centers to track the paths. This is because a sphere's 'center of gravity' tends to be at its center. Objects in 'free fall' will always tumble about their center of gravity.

If an object is irregular, the command SNAP CURSOR TO CENTROID may place the cursor near where the center of gravity would be in real life. This depends on the assumption that the vertices of the object are distributed the same as the real object's mass would be, which is often not correct.

Most people can learn to guess where an object's center of gravity should be. If you can pick the object up, just feel for its center of balance. In some cases, there is a rule of thumb to determine the center of gravity of an object. For example, on an airplane, the center of gravity usually lies about one third of the way from the leading edge to the trailing edge of the wing, except in unorthodox designs, like canards.

If the object is not in free fall, you might not want to place the local origin at the center of gravity. A robot arm, for instance, is often mounted to the floor. It would almost certainly have its local origin at its shoulder bearing. For another example, recall the slot car analogy. The slot car's

MOTION PATHS

guide pin is usually at the car's front end, well ahead of the center of gravity. This is why slot cars often 'fishtail' as they go through turns.

For realistic motion, you should always try to find a real life counterpart to the object you are modeling, and think of it in terms of its 'path', 'local origin', and 'tumble'. By examining reality, you can find the most effective way to devise the animated motions you want. If realism is not important, you can do just about anything you please!

By now you will have realized that what you see in the Tri-View as a static scene could represent a complicated animation. The next chapter will explain how to get from one to the other.

NOTES



Copyrighted material

CHAPTER 11

CREATING AN ANIMATION

In the preceding chapter we saw how to define a motion path in Sculpt Animate, but a motion path does not make an animation by itself.

For Sculpt Animate to construct an animation, it must go through an automated sequence of steps, beginning with one or more scenes which you create to describe the animation's movement, and ending, in most cases, with a compressed sequence of images that may be loaded into RAM and played back as an animation.

This chapter introduces a facility called a 'Take'. The Take is a control panel which lets you make choices about how the animation will be built.

This chapter will also explain how a few scenes, or only one, can be used by the Take to create the many images that an animation needs. At last you will be able to make your three dimensional objects move.

The Take

In the motion picture industry, a 'take' is a scene or portion of a scene that is photographed without interruption. Clearly a lot of organizational effort is required for a successful take.

In the world of Sculpt Animate, the Take is a special file which contains information needed to coordinate and control the construction of an animated sequence. For instance, it contains information about where each file associated with the Take is to be stored. It also stores data relating to the animation's imaging mode, screen size, etc., so that all of the scene's frames are rendered consistently.

ANIMATION

A number of other data are also kept in the take. Many of the Take's controls may be set directly by you. Other coordinating actions are performed by the Take automatically as you proceed. These actions and settings will all be explained in this chapter and the next.

Before you begin working on an animation, you must first specify the name of the Take. This name will be used to prefix all of the files that are created in association with this Take, so that different Takes cannot be confused.

MODIFY TAKE			
Take mode:		GLOBAL	
Number of frames:	10	RAW animation:	YES
Preview size:	FULL	Save images:	YES
Motion blur:	NO	Loop mode:	LOOP
Animation drawer:			
Foreground:			
Background:			
SAVE GLOBAL SCENE		LOAD GLOBAL SCENE	
PREVIEW		RENDER ALL	
		DELETE ALL IMAGES	
		OK	

Modify Take, GLOBAL mode

To set the Take's name, execute the command PROJECT LOAD TAKE. The familiar file requester will pop up. If you are continuing to work with a previously created Take, then select it in the normal way, from the list of files that is displayed, otherwise just type the new name into the appropriate slot. Click the OK gadget to continue with the new Take. If you change your mind about loading a new Take, click CANCEL to exit without doing anything.

You will find that you can still use Sculpt Animate just as before. You can load, edit, and save scenes as usual, and create and save images. As far as the normal use of the program is concerned, there is no difference.

ANIMATION

By naming the take, however, you have activated an extremely powerful command: EDIT MODIFY TAKE, which had previously been ghosted. Execute this command and look at the requester that pops up. The MODIFY TAKE requester is the control panel for making animations.

When the MODIFY TAKE requester comes up for the first time, it is in GLOBAL mode. At the top center of the requester, there is a toggle gadget which switches the requester between GLOBAL and its other mode, FRAME.

The screenshot shows a window titled "MODIFY TAKE". Inside, the "Take mode:" is set to "FRAME". Below it, "Current frame:" is set to "1". To the left, "Key frame:" is set to "NO". To the right, "Duration:" is set to "2". Below these are two empty text boxes for "Image drawer:" and "Scene drawer:". In the center is a filmstrip icon with a vertical line at frame 10. At the bottom are five buttons: "SAVE KEY FRAME SCENE", "LOAD KEY FRAME SCENE", "RENDER FRAME", "DELETE CURRENT IMAGE", and "OK".

Modify Take, FRAME mode

These two modes look radically different, because they permit you to deal with the animation in two entirely different ways. The GLOBAL mode contains settings and commands for the entire animation, while FRAME mode provides controls that let you deal with each frame of the animation individually, on a frame by frame basis.

To get a better grip on this distinction, examine each mode of the requester. First, set the mode toggle gadget back to the GLOBAL mode.

It should not be hard to imagine that each control that is visible in this mode has some function relating to the whole animation. For instance, the box marked 'Number of

ANIMATION

frames' is obviously where you type in how many images you want the program to create for your animation.

Choosing the best number of frames can be a complicated matter, involving the desired playing time, display duration of each frame, memory considerations, etc. For the sake of this discussion, type in the number 10.

If you click the GLOBAL/FRAME toggle gadget again, you can take a look at the requester's FRAME mode. In this mode, everything you do affects individual frames of the animation.

You should see a row of ten boxes stretched across the lower middle portion of the requester, looking suspiciously like a strip of photographic film. This is the film strip gadget. Each of these boxes represents a frame of the animation. By clicking in a box, you make that frame the 'current frame'. This will be shown by the number in the 'current frame' slot, and by a small arrow beneath the filmstrip.

Another control in this mode is the 'duration' setting. This is the display duration that we mentioned a moment ago. The number in the box is how long, in 'jiffies', the current frame's image will be displayed during playback. One 'jiffy' is 1/60th of a second. Longer frame durations make the animation seem slower. We will learn more of this later in this chapter and the next.

Before we continue, notice that you haven't seen any command to save the Take. All the data associated with the take is saved automatically whenever you click OK to get out of the MODIFY TAKE requester, and also when you start an animation render.

Two kinds of animation

Just as there are two different modes to the Take, GLOBAL and FRAME, there are two different ways you

can create motion in Sculpt Animate, global animation and key frame animation.

Global animation is specified with a single scene, called a 'global scene'. This scene contains motion paths and other objects that have their effect globally, affecting every frame of the animation. The global scene is also the best place to put objects that don't move at all, since this ensures that they will be present in every frame.

Key frame animation is specified by two or more scenes. A pair of key frames can contain objects that change position and shape from one key frame to the next. Key frames do not make any use of motion paths, and they only affect a particular group of frames.

As you might expect, creating global animations demands extensive use of the Take's GLOBAL mode, and key frame animations similarly are tied to the FRAME mode. There is nothing sacred about this linkage, however. Both the GLOBAL and FRAME modes provide tools that can be used for any animation.

Indeed, a particular Take may have motion and objects that are defined in the global scene, and others that are defined in key frame scenes. In creating the animation's images, these two separate forms of animation are essentially 'played back' together.

When the global and key frame animation methods are used together this way, one or the other may take precedence in determining certain placements and values. In general, any key frame scene that is defined will take precedence over the global scene in its World illumination and sky settings, as well as observer lens and tilt. The global scene's Location and Target placements can be forced to take precedence by entering their names in the global scene's hierarchy. This is explained more fully in the next chapter.

ANIMATION

Global Animation

You already have seen how to create scenes that contain objects which are attached to paths in a motion hierarchy. In essence, this is all there is to global animation, but in practice there is a bit more that you need to know.

In the Take requester, the two items that are essential to global animation are the buttons marked SAVE GLOBAL SCENE and LOAD GLOBAL SCENE, in the GLOBAL mode.

As you build a global animation, you piece together your objects and paths in a single scene. This scene may be saved and loaded just like any other, or pieced together from a number of other scenes. It is very much like building an animated machine of some sort.

You cannot make your scene's 'machine' run, however, until you load a Take and use SAVE GLOBAL SCENE in the MODIFY TAKE requester to save it. This tells Sculpt Animate to save the scene with a special file name associated with the Take. File names are discussed in the next chapter.

If you want to load in a previously saved global scene to work with, use the GLOBAL mode button marked LOAD GLOBAL SCENE, then click OK to get out of the Take requester. This command will clear the Tri-View automatically, so you don't need to execute an ERASE ALL first.

An example global animation

Enough explanation. It's time for you to make an animation. We have already mentioned orbital paths on something of a grand scale. For this animation, we will do something much simpler--make a cube fly around a circular path.

To start, clear the Tri-View with EDIT ERASE ALL. Execute PROJECT LOAD TAKE and enter a new name for the Take. Any name will do, try 'global'.

Now select the Down window and add a circle with 36 vertices. Select one vertex and indicate an adjacent vertex with the cursor. Now execute the command EDIT DO MAKE PATH. This will be our 'orbital' path.

Next, make a cube with the command EDIT ADD CUBE. Make sure you have deselected the vertex on the path and then select the cube. Call EDIT DO EXPAND and shrink the cube to a size that is fairly small compared with the circle.

People will usually want to grab the cube and place it 'on' the path somehow, but you don't need to worry about where the cube is placed. In the animation, it will automatically be shown on the path we attach it to.

We do need to set the cube's local origin, though. We'll do this in a moment, so leave the cube selected and move the cursor to its center with the command EDIT SNAP CURSOR TO CENTER.

Now we must set up a hierarchy. First, execute the command EDIT NAME SELECTED VERTICES. Enter the new name 'orbit', press return, and click on the OFFSPRING box. Now enter the name 'cube' and press return.

We have built a pretty simple hierarchy. The parent is called 'orbit', but nothing has that name yet, it is just holding a place in the hierarchy. 'orbit' has only one child, called 'cube'. Now click on the OK button to give that name to each vertex of the cube.

We will want the center of the cube to travel around the circle, so we must set a local origin for the cube. The cursor should still point to the center of the cube, so execute the command EDIT MODIFY LOCAL ORIGIN and click on the SET button. Leave the requester by clicking OK.

ANIMATION

Next we must name the path. Move the cursor to indicate one of the vertices on the circle and execute the command EDIT NAME INDICATED PATH. 'Cube' should be in the current slot where we left it. Click on the Parentage box, so that 'orbit' becomes the current name, then click on OK to leave the requester, giving the name 'orbit' to the circular path.

Our motion hierarchy is now complete, the cube is attached to the circular path. If you'd like, you can double check this by calling any hierarchy command, and seeing that 'orbit' names a path and 'cube' names vertices. Be careful, though. When you are just looking at the hierarchy remember to leave the requester by clicking CANCEL, or you might do something you didn't intend to.

Before rendering any animation, you must see that the observer's location and target are placed appropriately. Position the cursor above the plane of the orbit, at a distance of two or three times the size of the circle, then call OBSERVER LOCATION. Next position the cursor in the middle of the circle and call OBSERVER TARGET.

We have now specified the global scene completely, and we need to save it. While it is fine to save your interim work using SAVE SCENE, the finished global scene must be saved through the MODIFY TAKE requester

You have already given the take a name, so call up the EDIT MODIFY TAKE requester and click on SAVE GLOBAL SCENE. With the scene safely on disk, you can set other take controls.

Your final task is to set the number of frames. The path has 36 vertices, so click on the 'Number of frames' box, delete the previous number, type 36, and hit return. When you click OK, the take file will automatically be saved.

Congratulations! You have just prepared your first animation, now let us see how you can watch it move.

Previewing an animation

Later in this chapter, we'll see how to render an animation in full color, and compress the images for playback from RAM. This process can be time consuming, especially for large animations. In this particular case it would be unrewarding as well, since we've neglected to put a lamp in the scene.

Right now, however, all we need to see is the motion of the scene, and we aren't particularly interested in keeping the results for posterity. For this purpose, Sculpt Animate provides quick preview facility. This facility uses WIRE FRAME mode in low resolution, so it can assemble the animation in a minimum of time and memory.

To start the preview, execute the EDIT MODIFY TAKE command and make sure that the Take mode is set to GLOBAL. Then click the PREVIEW gadget. Each frame of the animation will be displayed as it is calculated. When all the frames are done, the entire animation will be played repeatedly.

The initial speed (frame rate) of the playback will depend on the duration that you have set for each frame of the animation. In our example, we haven't changed the default setting of 2 jiffies per frame, this means each image will be shown on the screen for 1/30th of a second (1/25th of a second on PAL Amigas).

ANIMATION

While the preview is being played, you can press the keyboard's top-row digit keys 0 through 9 to modify the speed as follows:

Key	Playback Speed
0	normal rate
2	2 jiffies per frame
3	4 jiffies per frame
4	8 jiffies per frame
5	16 jiffies per frame
6	30 jiffies per frame
7	60 jiffies per frame
8	120 jiffies per frame
9	240 jiffies per frame

Pressing the spacebar will momentarily halt the playback, pressing again resumes it.

When you want to stop the preview entirely, press the right mouse button or the ESC key. The preview screen will be closed, and you will be asked whether you want to unload the preview.

If you think you will want to view the preview again later in the session, you can answer NO to this question. This causes the preview to be kept in memory. It can be replayed from the Tri-View by executing the command PROJECT SHOW PREVIEW.

While the preview data is retained in memory, you will have less memory for other things. You can free up this memory at any time by executing the command PROJECT UNLOAD PREVIEW.

If your animation is very long, the preview display may fail because of insufficient memory. In that case you can return to the MODIFY TAKE requester, select GLOBAL mode, change the 'preview size' gadget from FULL to HALF, and try the preview again. Because the image

generated has one fourth the area, the preview will only take about one fourth the memory.

Never hesitate to use the preview facility, it is much faster than generating a full animation, and it will often show you ways to improve your object motion and placement, as well as your observer's position. The time you save will be well worth this extra step.

Looking at individual frames

When you are working with a global scene, objects and paths will be placed however they were constructed. Objects that are to follow paths in the animation might be placed in an entirely different part of the scene as you construct it. In the global scene this is perfectly all right, since an animated object will move to the active vertex of its path anyway.

As Sculpt Animate generates the images of the animation, it will resolve the relationships between objects and paths for every frame, creating a sort of 'working version' of the original scene.

If you want to see what this 'working version' of the scene looks like in a particular frame, call the MODIFY TAKE requester and click to the FRAME mode. Select any frame of interest, click LOAD KEY FRAME SCENE. This command automatically clears the Tri-View, so you don't need to execute an ERASE ALL first. Now click OK to get back to the Tri-View.

What you will see is the scene that Sculpt Animate will use to create this frame's image. You can now move and resize the Tri-View to examine various parts of the scene. This allows you to make sure that the animation is proceeding precisely as you expected.

When you are looking at an individual frame's working scene, you will find that some Tri-View commands will not work as you may expect. For instance, none of the selection or editing functions will work on the global scene's objects.

ANIMATION

The next section will explain that you are actually in a mode which allows you to edit key frame objects while seeing how they relate to global objects. You just don't have any key frame objects yet.

If you want to render an image of a frame's working scene, select that frame in the FRAME mode of MODIFY TAKE and then click RENDER FRAME. The image will be rendered in the size and imaging mode that currently applies for the take. This will be what was set for the last global or key frame scene that was saved before the last DELETE ALL IMAGES command. The full effect of these commands will be explained later.

If you want RENDER FRAME to match the preview images, set the imaging mode to LO-RES, NO-INTERLACE, WIRE FRAME.

Key frame animation

We mentioned in Chapter 1 that there are two ways of animating objects: global animation and key frame animation. We have already seen global animation in action. It depends on a single scene, the GLOBAL scene, which describes objects and motion that are to exist throughout the animation. It works like a machine, in which groups of vertices run about on tracks.

Key frame animation works on a much different, somewhat simpler principle. The idea is this: take two scenes that are identical, then change the position of some vertices in one of the scenes. The two scenes now describe a motion for those vertices.

If this kind of animation is to work, then the scenes that describe the motion must be linked to particular frames. These frames are called 'key' frames. The scenes we save for these frames are called key frame scenes.

You have to tell Sculpt Animate which frames will be key frames. This is done in the FRAME mode of the

MODIFY TAKE requester. You simply select the frame you want, then click the 'Key Frame' toggle gadget from NO to YES.

Once you have tagged the frame as a key frame, you can click SAVE KEY FRAME SCENE. This saves whatever scene is in the Tri-View as the current frame's scene.

A key frame scene is created much like any other, by editing in the Tri-View. The contents of these scenes must be coordinated in a particular fashion, which relates to how Sculpt Animate uses the key frame scenes.

We have said that certain frames will be key frames, so it follows that not every frame needs to be a key frame. The position of key frame objects in non-key frames is derived by comparing the two adjacent key frames, one before and one after the current frame. Specific rules are applied which will be explained later.

Key frame motion is resolved when Sculpt animate constructs a 'working scene' for the frame that is being rendered. This is the same working scene that we mentioned with respect to global animation. If there is a global scene, it will also be resolved and contribute to the frame's image.

Even if all of the motion you design is done in key frames, it is likely that your animation has some components that do not change in any of the frames. These may be stationary objects like furniture or trees. Objects like this may be placed in the global scene. By keeping these objects in the global scene, you avoid having to worry about them in the key frame scenes, and you reduce the amount of storage required for the Take.

As we saw before, you can use the MODIFY TAKE gadget LOAD KEY FRAME SCENE to load in a 'working scene' representing the state of the animation in the current frame. This command automatically clears the Tri-View, so you don't need to execute an ERASE ALL first. Click OK to get back to the Tri-View.

ANIMATION

If there are objects in the global scene, they will now be visible in the Tri-View, but they may not be edited or changed. If the frame is not a key frame, objects present in the adjacent key frames will also be visible, but not alterable.

If, however, you have selected a key frame, that frame's scene will be loaded in. Global objects will still be visible so that you can use them as a reference for placing key frame objects. When you have finished editing, call **MODIFY TAKE** again, and click **SAVE KEY FRAME SCENE**. Of course, only the key frame objects will be saved.

We will discuss additional key frame editing techniques later in this chapter.

In-betweening

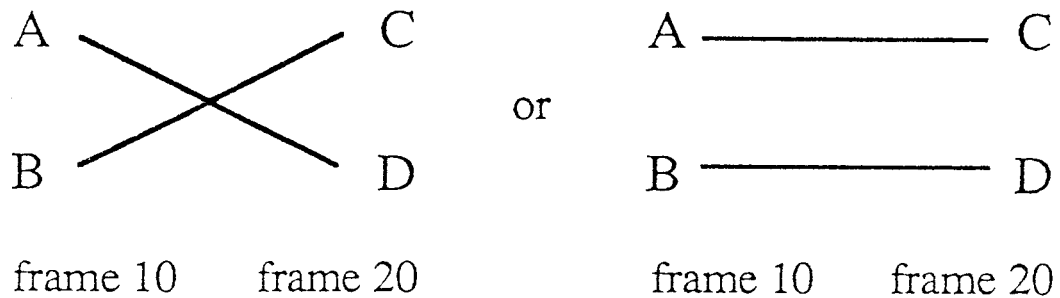
We have said that Sculpt Animate 'resolves' the difference between two adjacent key frames, creating motion. This deserves more explanation.

Suppose that you specify frame ten and frame twenty to be key frames, with no other key frames in that interval. Each key frame has its own scene. When Sculpt Animate comes to render an image for a frame that is not a key frame, frame thirteen, for example, it interpolates between the key frames that lie to either side.

Suppose that each key frame contained only a single vertex, then the interpolation process, or in-betweening, would assume that the vertex moved smoothly from its position in frame ten to its position in frame twenty. In frame thirteen, the vertex is placed 30% of the way along the path.

So far, so good. Now let us complicate the scenes and place two vertices in each key frame. The in-betweening process can be carried out as before, provided that we know which vertex in frame ten corresponds to which vertex in frame twenty. It is up to you to supply the clues that let Sculpt Animate resolve the dilemma. Suppose the two

vertices in frame ten are called A and B, and the two vertices in frame twenty are called C and D, then which of the following do you want?



The second might seem preferable, in that the paths do not cross, but in three dimensions, the majority of lines do not intersect. If we had more than two vertices, it would become still harder to decide which vertex in one frame corresponded to a vertex in the other frame.

You could be required to assign a unique name for each vertex, but that would be a painful chore if your scene had a thousand or more vertices. Fortunately, this is not necessary.

It turns out that the vertices and other parts of a scene are stored in a particular order as they are created. This order remains the same in the scene as long as nothing is added or deleted. In particular, the order remains the same whenever vertices are moved.

In order for key frame animation to work, you need only make sure that consecutive key frame scenes are modified copies of the same scene.

If this sounds complicated, it could not be easier in practice. If there is to be a global scene, you can design it first and save it with `SAVE GLOBAL SCENE`, this can be helpful in editing key frames, as we mentioned.

Now, suppose that our animation has one key frame every five frames, so that the key frames are 1, 6, 11, 16 etc. In the `MODIFY TAKE` requester, go through and mark every fifth frame as a key frame. Now use `LOAD KEY`

ANIMATION

FRAME SCENE to load frame number one. There is no key frame data yet, so only the global portion of the frame will be visible in the Tri-View. Assemble the scene using any of the regular Sculpt Animate tools, and save the scene with SAVE KEY FRAME SCENE when you are finished.

Scene number one has been saved now, but a copy of it is still in the Tri-View. You can manipulate this scene, move or rotate objects, apply the magnet tools, do anything that you wish, but do not create or erase any vertices, edges or faces.

If you somehow alter the scene's structure, say by accidentally adding a vertex, just use LOAD KEY FRAME SCENE to load in the first frame again.

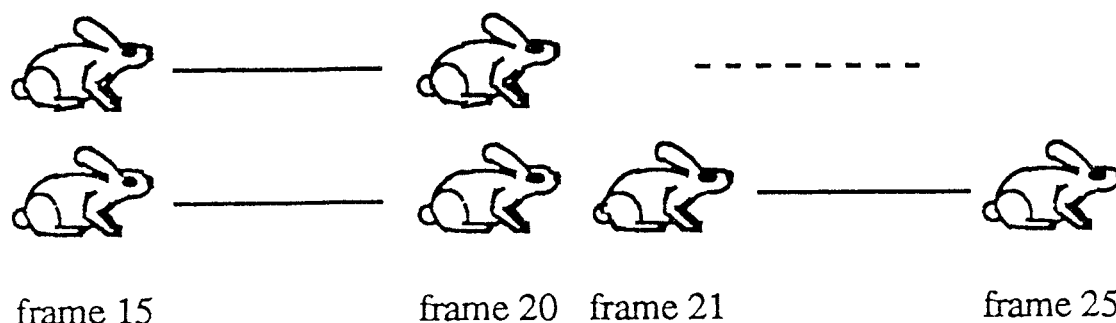
When the scene is to your liking, go back to MODIFY TAKE and change the frame number to six, then use SAVE KEY FRAME SCENE to save the scene. Sculpt Animate can now interpolate between frames one and six with no danger of ambiguity, since each vertex, edge and face in key frame one has its matched twin in frame six.

You can apply this technique to bridge the gap between any two key frames that are separated by a number of non-key frames. There is a quick way to check a pair of key frames to make sure their structures match. Just select a frame between them and call LOAD KEY FRAME SCENE. If the structures don't match, an error message will appear. Otherwise, the interpolated scene will be loaded as usual.

By now you may think that all of a Take's key frame scenes must have the same structure, the same number of vertices, etc. Fortunately, this is not always the case. This strict rule permits in-betweening, but if two adjacent frames are both key frames, no in-betweening takes place. Two key frames placed together may be as different as you like.

Suppose that you have two animated rabbits, and you want one of them to magically vanish after frame 20. Build the scenes as usual for the key frames before frame number 20. Make sure that frame number 20 is a key frame, and

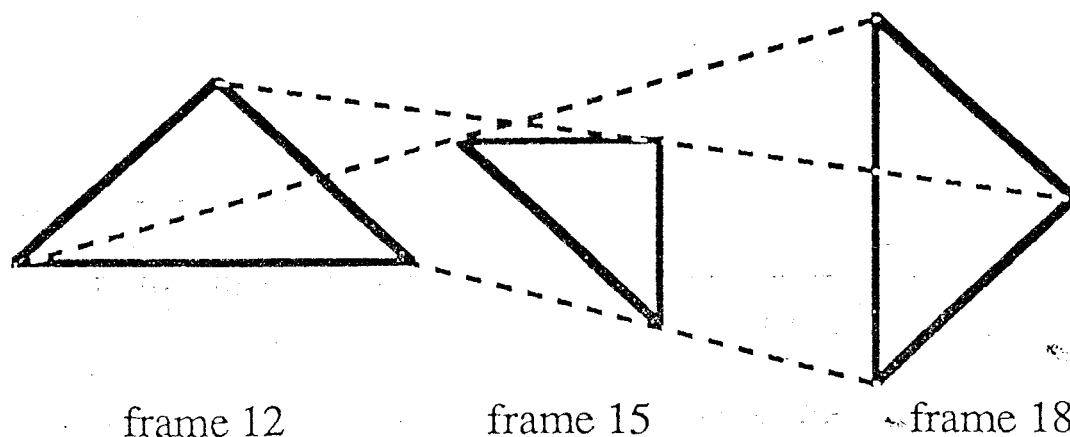
save its scene. Now make frame number 21 a key frame also. Erase one of the rabbits in this frame's scene and save it, then continue to construct subsequent key frames, ensuring that their structures match as needed.



The rule is simple, whenever you have a frame that is not a key frame, the key frame before and the key frame after must be derived from the same scene, with nothing added or removed. Whenever additions or removals are required, you must use two adjacent key frames to make the transition.

In-betweening distortion

The in-betweening process is very simple, involving linear interpolation, but it can produce strange effects. Consider the following case: two key frames, frames number 12 and 18, with no key frames between them. A triangle in frame 18 is identical to the corresponding triangle in frame 12, except that it has been rotated by a moderate angle, say 90 degrees. The in-betweening process for frame 15 is shown below



ANIMATION

You can see that at frame 15, the triangle generated by the in-betweening process is much smaller than the two identically shaped triangles in the key frames. The way to minimize this distortion is to insert more key frames, and ensure that objects do not change too much from one key frame to the next.

An example key frame animation

Let us remake our animation of the orbiting cube using key frame animation. First clear the Tri-View with EDIT ERASE ALL. If you have been working with a Take, you will be asked if you wish to erase it. Answer OK to this-- nothing will be lost, the Take still exists safely on disk, you are just erasing the copy that is held in memory.

Next, add a cube to the Tri-View and click the Expand Tri-View gadget until the cube is about one eighth of the size of the Tri-View. Select all the vertices and position the cursor just inside the Down window's bottom edge, in the middle.

This will be the animation's center of interest, so execute the OBSERVER TARGET command, to place the target.

We will be flying the cube around the target in a circular path, so we must set the observer where it can see the action. Expand the Tri-View again, this time until the distance from the target to the cube is about one eighth of the Tri-View's width, and place the cursor at the bottom of the Down window. Go to the North window and move the cursor up by about one third of the height of the Tri-View. Execute the command OBSERVER LOCATION.

Place the cursor on the target--this will require moving it in two windows. Next click on the Center Tri-View gadget, then click Shrink Tri-View until the cube is just about to leave the Tri-View.

When you 'zoom in' with Shrink Tri-View, the cursor and target might seem to drift apart. This is because

shrinking the Tri-View increases its resolution, making a small error in placement seem larger. If this happens, just move the cursor back to the target again.

Now we are ready to set the cube in motion. Execute PROJECT LOAD TAKE to name a new take (try 'key_frame'), then call up the MODIFY TAKE requester and enter 36 as the number of frames in the global mode. Remember that you need to click in the box, change the number, and press the return key to enter the new value.

Since our cube is to fly around a full circle in 36 frames, it must move 10 degrees per frame. Let make every fourth frame a key frame. Click on the Take mode gadget so that it becomes FRAME. Click on the film strip gadget for frame number 1 and click on the Key frame box so that it changes from NO to YES. A letter 'K' will appear in the first frame of the film strip image to indicate that this is a key frame. Now click on the fifth frame and again click on the Key frame box. Repeat the process until frames 1, 5, 9, 13, 17, 21, 25, 29, 33 and 36 are all key frames.

It doesn't take a mathematician to see that 36 is only the third frame past 33, but there's a good reason for this inconsistency. To see why, recall that in the global animation we also used 36 frames. The circular path naturally had a 10 degree arc between vertices. Of course, one of these arcs was the one linking frame 36 and frame 1. This arc had its effect when the animation looped, making a smooth transition from one cycle to the next. Our key frames don't specifically account for this last 10 degrees, but if we just leave off 10 degrees 'short' at frame 36, everything will work when the animation loops. Honest!

Examples are for doing, so let's create the scenes. Click on the first frame in the film strip gadget, then click on the SAVE KEY FRAME SCENE button. This saves the scene in the Tri-View as the first key frame. Now we must construct the next key frame scene, which is frame 5. Click on the OK gadget in order to leave the requester.

ANIMATION

Back in the Tri-View, make sure that the cursor is lined up over the target, and that the cube is visible near the top of the down window. We need to rotate the cube around the target in a clockwise direction by 40 degrees. You can either click on the clockwise rotation gadget in the Down window 8 times, or click on it once with the right Amiga key held down (for a 45 degree rotation) and then click the counter-clockwise gadget in the Down window once (to back up by five degrees).

Return to the Modify Take requester with an EDIT MODIFY TAKE command. Click on the image for frame number 5 and press the button SAVE KEY FRAME SCENE.

Repeat this process for each frame, rotating the cube around the target and saving the key frame scene. Remember we need to leave frame 36 short by 10 degrees, so only rotate the cube 30 degrees (6 clicks) for this frame.

Once you have saved all the key frames, all you need to do to see your handiwork is toggle to the GLOBAL mode and click PREVIEW.

Global vs. Key frame

If you have followed through with this chapter's two less-than-exciting animations, you may have noticed a couple of differences. The circular motion in the global animation looks smoother than that of the key frame animation. This is because, due to in-betweening, the actual path of the cube in the key frame was a nonagon, or nine sided figure, while in the global animation it was a 36 sided figure, which is much closer to a circle. Take care, when using key frame animation, that the interval between key frames is not too large, or motions may become unacceptably jerky.

The other difference is more obvious. In the key frame animation, the cube can be rotated so as to keep the same face towards the center of the orbit, like the moon does to the

earth, while the cube in the global animation did not rotate at all.

To make the cube rotate in the global animation, you must rotate the tumble axes. To see how, do an ERASE ALL to clear out the key frame Take, then call PROJECT LOAD TAKE. This time, find the name you gave the global example then double-click that name in the list of files.

Invoke the Modify Take requester and click on the LOAD GLOBAL SCENE box. Now leave the requester and point the cursor at the most easterly vertex on the circle and execute the command EDIT MODIFY INDICATED TUMBLE. Move the requester so that you can see the down window. Click on the Method box, so that it changes from INTERPOLATED to SPECIFIED.

This will do for the first vertex. Now click the NEXT button four times and to move the tumble axes around the orbit. Click the Method box to set the value to SPECIFIED. Rotate the tumble axis by 40 degrees, and the white tumble axis will move to become tangent to the circle again.

Continue around the circle in this fashion, setting the tumble axes appropriately for every fourth vertex. As you proceed, of course, the interpolation mechanism will be trying to help, so use the MATCH LAST gadget before setting each vertex. This ensures that you rotate its axes exactly 40 degrees. For most 'spinning' paths, a setting is needed every 30-45 degrees for interpolation to be smooth.

You can probably see that each of these types of animation has its own strengths. Global animation is able to describe smooth, flowing motions with complicated tumbling, free of key frame distortion. On the other hand, key framing allows things that global animation doesn't, such as changing an object's shape or color, and its an easy way to describe simple point-to-point motion.

Most people will 'take' to one method or the other at first. This is fine, explore the one that makes the most sense to you right now. The other will wait patiently until you've

ANIMATION

gained confidence. Eventually you will be able to pick the one that best suits whatever object or motion you have in mind.

RAM animation

So far, all of the animations you have done were WIRE FRAME previews. These two-color images are small enough in memory that they can be stored directly in RAM and 'page-flipped' to the screen for animation. We've seen that Sculpt Animate can create much more beautiful images than these, in full color and much higher resolution. The more impressive images have a cost, though, in the amount of data they contain.

An image rendered in HI-RES, INTERLACE, PAINTING, for instance, can take over 16 times the memory that a preview image does. Typically, color images occupy between 50,000 and 100,000 bytes. To show a single second of animation, then, involves moving about 2 megabytes of data into the Amiga's 'chip' graphics RAM every second. This data rate is more than the Amiga's processor can handle, even if the whole animation could be loaded into expansion memory and copied from there.

Fortunately, in most cases, not all of the image data needs to be moved. One frame in an animation is usually very similar to its predecessor, so only the differences between the frames need to be changed. This technique was used to produce the Juggler demonstration in 1986, and has since become known as 'delta' compression. (The Greek letter delta is often used to symbolize change or displacement.) The data describing the changes from one frame to another is called 'delta data'.

In delta compression, an animation is reduced to an initial image plus a series of data chunks that can be used to reproduce each subsequent image by altering the current one. A standard file format, called the ANIM format, has been suggested for animation files. Sculpt Animate employs a variation of this format called J-type animation. At this

writing, at least three other ANIM formats are in use. A format converter for use with Sculpt Animate anim files is pending.

RAM animation is not without its drawbacks. Because two consecutive images are compared on a pixel by pixel basis, some animations may not be amenable to delta compression. The problem occurs if a very large part of the image changes from one frame to the next, as when large objects pass very close to the camera.

Even more extreme cases are when the observer's location or target are moved, the observer lens's focal length is changed, or lamps are moved. These are advanced techniques that will be discussed in the next chapter. Each of them virtually guarantees that nearly every pixel of the image will change. Since some overhead is involved in compression, this can lead to 'compressed' anim files that are larger than the original image data.

Too much motion will not only cause extremely large anim files. It also slows animation playback, since it increases the amount of data that must be moved by the processor. In a typical application, if you do not achieve a compression ratio of at least three-to-one between the image size and the delta data, the animation is unlikely to replay at the desirable rate of two jiffies per frame (25 or 30 frames per second).

The other obvious restriction for RAM animations is that the amount of memory on your machine limits the total duration of the animation. If each image is moderately different from the next, you can expect to play about three seconds of animation for each megabyte of available memory. If only small portions of an image are changing, the duration could be as long as 30 seconds per megabyte.

The apparent length of an animation can be increased by designing it in such a way that it repeats itself in a short length of time. Such looping animations are obviously less interesting than a full length feature film, nevertheless, impressive animation demos have been made in this way.

ANIMATION

You need not restrict yourself to a single object moving in a circle. For example, you could produce a scene of a column of troops marching along a twisting road. In the time that it takes on figure to march the distance between himself and the person in front of him, the scene can be made to exactly repeat itself. An animation can also be extended by carefully placing pauses in the action, in the form of particular frames with long duration values. Although RAM animation does impose some severe restrictions, there is no limit to the creativity that can be applied to this new art form.

RAM animations serve two purposes. As a stand alone end product they may be acceptable for short animations, a spinning logo for a commercial, for example. Alternatively, short animation Takes may be assembled into longer sequences by combining them on video tape using standard videotape editing techniques. This probably represents the most cost-effective approach to computer generated video animations.

Making a RAM animation

The button marked RENDER ALL in the GLOBAL mode of the MODIFY TAKE requester starts the rendering. The preparation required for this process is essentially identical to that for a preview, but there are a few additional items that need to be attended to.

First, you obviously need to consider color, texture, and lighting in preparing your animation. The examples created so far didn't even have any lamps, you should place at least one in the global scene. Everything you have learned about creating good images applies here. Don't fail to try a few test shots with the faster modes and sizes before committing to the full render. Look for good color, lighting, and composition. Remember, though, that the fancier textures look pretty bad in SKETCH and PAINTING; take this into account.

You will also need to remember to set the imaging mode and size for your animation. These settings are taken automatically from the last scene saved with either SAVE KEY FRAME SCENE or SAVE GLOBAL SCENE.

An animation may consist of dozens or even hundreds of images, and take days to render. Because of this, be sure to render a few test images here and there throughout the animation, using the RENDER FRAME button in the FRAME mode of MODIFY TAKE. This command will render the frame's image in the current imaging mode, as discussed before, and then save the image to disk.

Finally, it is a good idea to click DELETE ALL IMAGES before starting a full render. If you have used RENDER FRAME in setting up your animation, then some undesirable images may remain on disk. These will not be re-rendered. Also, the color map and imaging mode may have been 'locked'. The meaning of this will be explained in the next chapter, but for now just remember that DELETE ALL IMAGES resets the Take and makes it ready to start a new render.

If you are sure that the color map and imaging mode are OK then you can skip the DELETE ALL IMAGES. This avoids having to render those images twice. If you want to delete one image, but not all of them, use DELETE CURRENT IMAGE in the FRAME mode.

When you finally click RENDER ALL, a process will begin that closely resembles the PREVIEW render, with a few exceptions. We will look at this process as it would occur normally, with the GLOBAL mode controls (except number of frames) set to their defaults. We'll discuss variations in the next chapter.

First, Sculpt Animate will go through the animation frame by frame, rendering all the images. These images will be saved out to disk individually, by default in the same directory that was specified in the LOAD TAKE requester. Unless the animation is fairly small in size, you will need a

ANIMATION

hard disk to handle all this data en masse. Alternatives for floppy disk users are discussed in the next chapter.

Once all the images have been rendered, Sculpt Animate will again proceed through the animation. This time it will go through leapfrog fashion, finding the differences between alternating pairs of images. It starts as you might expect, with frames 1 and 2, but then it goes to 1&3, 2&4, etc. The images are compressed this way to facilitate 'double-buffered' playback. Double buffering keeps two images in memory for the best possible speed, one is updated while the other is displayed.

As each delta is found, it is written out to the animation load file, or 'anim' file. This file will contain the complete compressed animation, ready to be loaded and played back. The anim file is also saved to the Take directory by default. Consult the next chapter if this is not desired.

The compression phase can be quite lengthy, since the compression scheme was designed for greatest compression and fastest playback speed, not shortest compression time. You can use a much faster compression scheme by clicking MODIFY TAKE's RAM animation gadget from YES to ECONOMY, its second state. ECONOMY compression is much faster in calculation, but generally creates larger anim files that may not play as fast.

When the animation render is finished, you can click OK to get back to the Tri-View. If you have to interrupt the render (by pressing CTRL-C), or if it is stopped for any other reason, any images that have been saved will still be present, and will not need to be re-rendered. When you click RENDER ALL again, the render will pick up where it left off. If the compression phase is interrupted, it will start again at the beginning.

Playing a RAM animation

Before an animation can be displayed, it must be loaded into memory. This is done with the command PROJECT

ANIMATION

LOAD ANIMATION. A familiar looking file requester will enable you to select which animation you wish to play. Having loaded the animation, it can be displayed with the command **PROJECT SHOW ANIMATION**. The animation can be stopped by clicking with the right mouse button or by pressing the ESC key, The speed can be changed in the same way that the speed of a preview is changed.

After stopping an animation, it remains in memory so that it can be replayed without having to be reloaded. Use the command **PROJECT UNLOAD ANIMATION** to make the memory available for other purposes.

One imaging mode, **HI-RES INTERLACE** (non HAM), requires more chip memory for playback, in the larger image sizes, than is available in the Amiga while **Sculpt Animate** is running. If **Sculpt Animate** won't play an animation, this may be why. The freely distributable **Movie** program, included on one of your distribution disks, can play any **Sculpt Animate** animation.

NOTES

CHAPTER 12

ADVANCED ANIMATION TECHNIQUES

The last chapter walked you through the basics of animating scenes with Sculpt Animate. The discussion has focused on the simplest conditions, leaving many of the available options in their 'default' state. This chapter describes these additional options, which you can use as you gain confidence in creating animations.

Also in this chapter are sections which introduce some advanced animation techniques.

Loop mode

One of the options we have ignored is provided by the MODIFY TAKE gadget marked 'Loop mode:'. This option only applies to RAM animation and PREVIEW renders. For other animation types the gadget will not be visible.

The default setting for this gadget is LOOP. We have already seen this mode in action. In it, the animation starts over again after the last frame is displayed. We have also seen that the last and first frames should usually not be identical, otherwise the motion will seem to pause for an extra frame when the animation loops. A five frame animation would loop as 1, 2, 3, 4, 5, 1, 2, 3, etc. When LOOP is in effect, RAM compression will not stop at the last frame, since deltas must be calculated to carry the animation through to frames 1 and 2.

An alternative way of producing an animation is to have it replay in the form of an oscillation. Set the loop mode gadget to OSCILLATE before rendering the animation. A five frame oscillating animation will play the frames in the following order: 1, 2, 3, 4, 5, 4, 3, 2, 1, 2, 3 etc.

ADVANCED TECHNIQUES

If the loop mode gadget is set to NONE, then the animation will simply play to the end and then stop.

Frame duration

We have already mentioned frame duration, but it is important enough that it deserves a fuller explanation.

The 'duration' of a frame is the time, in jiffies, that that frame's image will be displayed during playback. The duration might actually be longer than what you set, since some animations can be too complex to play back at full speed.

The definition of 'jiffy' can vary, depending on where you live. In the United States and Japan the video standard is NTSC, which displays each image as two interleaved fields, at 60 fields per second, so an NTSC jiffy is 1/60th of a second. In Europe, Australia, and elsewhere, the video standard is PAL. A PAL video jiffy is 1/50th of a second.

If the images are being sent to a single frame controller (explained later in this chapter), then the set duration for each frame will be passed to the driver. The frame controller driver must decide how to interpret this value; for instance, a video frame controller might record a field for every jiffy, a film recorder might do otherwise. The driver's documentation should explain how it uses this data.

When you first call up the FRAME mode of the MODIFY TAKE requester, you will see that the duration of frame one is shown as zero. When a frame's duration is set to zero, this has a special meaning. The duration actually used will be the same as that used for the frame before; for frame one, this duration is assumed to be two. Since the durations for a take are all initialized to zero, the default frame duration for the whole animation is two jiffies, or 1/30th of a second in NTSC.

If you click to subsequent frames, you will see that the duration is shown as '(2)'. The parentheses show that this

ADVANCED TECHNIQUES

is a derived value. When you click back to frame one, its duration is now shown this way, too.

This odd behavior allows you to set the duration for a large number of frames at once. If you click in the duration box and delete the old value (be sure to delete parentheses, too), then you can type in a new number. This number will take effect for this frame, and all subsequent frames that have not been set otherwise.

One note of caution: If you want to set a unique duration for a frame somewhere in the middle of the animation, you will need to set the next frame's duration back to the default. Otherwise the rest of the animation would take on the new value.

Frame duration is directly linked to both the apparent smoothness and the playing time of the animation. In NTSC, a 30 frame animation will last only one second at two jiffies per frame.

You will find that two jiffies is a very short time. Path motion that is to be played at this rate will probably seem overly fast unless the path used is somewhat densely subdivided. This means more frames overall, and longer rendering time for the animation.

If the ultimate goal is a pleasant representation of the motion, then the higher rate may be necessary. In this case, it might be wise to stick to the PAINTING or SNAPSHOT modes, and avoid PHOTO if rendering time is critical. Indeed, if motion is the primary subject of the animation, even WIRE FRAME can do very well at conveying this.

If a positional representation is all that is desired, and stunning imagery is to be the centerpiece of the work, you can specify durations of four or more jiffies and still get the point across. This drastically reduces the number of frames, and hence rendering time, at the cost of some jerkiness in the motion. With practice, you will learn how to weigh these considerations according to your own taste.

ADVANCED TECHNIQUES

File name conventions

If you use the Amiga's command line interface (CLI), you will probably have noticed that Sculpt Animate creates image files with names that end in '.image', and object and scene files that end in '.scene'. This is how the program is able to tell different types of files apart.

You will not be surprised to learn that a Take file ends in '.take'. The Take file is quite small, because it does not actually contain any scenes or data, instead it accesses separate scene and data files.

When you name a Take, you are also indirectly naming all of its scenes. If the name of the Take is 'xyz', its file name is actually 'xyz.take'. The take's global scene file is called 'xyz.g.scene' and the first key frame scene is called 'xyz.1.scene'. If frame 20 is a key frame, its scene would be called 'xyz.20.scene'. When images are generated, they have names that are similar. The name for the image for frame number 20 is 'xyz.20.image'.

This simple naming convention is carried out automatically, so you don't have to worry about it. When you are using the Amiga's CLI to maintain your directories, however, it makes it easier to distinguish between the files of various takes. The Amiga's LIST and DELETE commands permit the use of patterns, for instance the command 'LIST p xyz.#?' will list every file associated with the Take named 'xyz'.

Note that Sculpt Animate does not save '.info' files for the files it creates, so they will have no icons on the Workbench screen. Sculpt Animate can easily create hundreds of files on a single directory, so icons would quickly become useless. The '.info' files themselves would effectively double the number of files, drastically slowing disk access for some operations.

If you haven't done so already, you will find it valuable to study your Amiga's CLI documentation before going very far with Sculpt Animate. The definitive source for this

information is the The AmigaDOS Manual, from Bantam Books, which is available through most bookstores where computer books are sold. This book fully explains the Amiga's file system and CLI. There are also several novice-oriented AmigaDOS texts on the market.

Disk space

In the course of generating an animation, you may find yourself dealing with many megabytes of data, so a little advance planning is wise. At times, you may need to estimate how much disk space each image will take.

Sculpt Animate's images are stored in a compressed format, which can vary in size from image to image, so you will need to check the size of an image from your particular animation. To do this, use `RENDER FRAME` to generate a test image before you click `RENDER ALL`. Try to pick the 'busiest' frame--an image with many transitions from one color to another will generally use more disk space, and be a 'safer' indicator of what's needed.

Once `RENDER FRAME` has saved the image, go back to the Workbench screen and the Amiga's CLI window. Use the `CD` command to change your current directory to the drawer where the image is stored (the same drawer where you opened your Take). Then use the AmigaDOS `LIST` command to see how many bytes are in the test image.

In the Amiga's operating system, disk storage is measured in 'blocks' of 512 bytes. Typically, because of system overhead, only 488 bytes will be useable per block. A newly formatted Amiga floppy disk can hold 1751 blocks of data (1758 if the 'noicons' option is used).

To see how many blocks are used by your test image, divide its number of bytes by 488, round up, and add 1. To see how many images will fit on a floppy, divide the number of blocks into 1751, round down, then subtract one to be safe.

ADVANCED TECHNIQUES

If the floppy is partially full, or you will be using a directory on a hard disk or some other device, the CLI "INFO" command will show how many free blocks remain on each device in the system.

Organizing floppy disks

So far, our discussion applies to using Sculpt Animate on hard disk systems as well as floppies. While the hard disk is preferable, by far, it is possible to render and compress RAM animations using only floppy disks.

The key to this endeavor is the old maxim: a place for each thing, and each thing in its place. You will need to find room for the Take and its associated scenes and images, as well as a place for the final animation.

Sculpt Animate makes use of 'overlays', segments of code that are loaded in at various times, depending on what is being done. This reduces the amount of memory required to run the program, but it means that the program will need to consult its own disk at times during a session. Sculpt Animate also uses code libraries kept on the Workbench disk. To reduce the amount of swapping required, you can 'boot' from your working copy of the Sculpt Animate disk (insert it instead of Workbench).

The Take and scenes will usually fit on the floppy disk with Sculpt Animate, so they're no problem. If the scene files need to be placed on separate disks, the procedure will be much the same as for image files, which we will discuss in detail.

Image files definitely are a problem, since as few as five might fill up a disk. A full disk error will bring the rendering process to a grinding halt. The anim file is also a problem, but we'll discuss that in a moment.

To avoid running out of space while rendering, use the last section's formulae to figure out how many of your animation's images will fit on a disk. Before you start the

animation, format (initialize) enough disks to hold all of the frames' images. Name each of these disks with a short, unique name, like 'images1-5', 'images6-11', etc.

Now, how is Sculpt Animate to know which image goes on which disk? You are able to specify a disk name in the 'image drawer' gadget of MODIFY TAKE, in the FRAME mode. Select each frame in turn, and type in a disk name for every frame. Use the 'scene drawer' gadget if you need to do this for key frame scenes.

Do not type in 'df0:', or any other device names, into these drawer gadgets. This does not give Sculpt Animate any way to ask for the correct disk. Use only the unique volume names of one of the floppies you have formatted. Remember, however, that a disk's volume name must be entered into the gadget with a trailing colon, e.g., 'images1-5:'.

The image and scene drawer gadgets work similarly to the duration gadget, in that any name you enter will be carried over to any subsequent frames that don't have an image drawer specified. This reduces the amount of typing that has to be done. A derived name (one that is carried over) will be shown in parentheses. Be sure that you erase the parentheses whenever you enter a new value.

As Sculpt Animate renders the frames it will ask for the next disk by name when the time comes to write an image to that disk.

When all the frames have been rendered, Sculpt Animate will begin to compress the image data. This is when the anim file must be opened. It is best to have a blank floppy disk dedicated to the anim file, since it can be very large. This disk should be formatted beforehand, and its name entered in the 'animation drawer' gadget of MODIFY TAKE, GLOBAL mode.

As the anim file is constructed, Sculpt Animate will again ask to see each of the image disks in turn. If you have only one floppy drive, this will undoubtedly incur a painful

ADVANCED TECHNIQUES

session of disk swapping, since the anim disk will also be needed for each frame. If you have two floppy drives, you can generally place the anim disk in one drive, then cycle the image disks through the other. This avoids much, but not all, of the disk swapping.

There is a practical limit to the size of an anim file: about 850K. This is the maximum size of any file on an Amiga floppy disk. It is possible that larger animation files may be assembled on hard disks, but they must be split into portions to be transported on floppies, public domain split programs exist which will perform this function. The AmigaDOS JOIN function can be used to re-assemble a split anim file.

Saving images

You can reduce a great deal of the trouble of working with floppies if you set the 'save images' gadget in the MODIFY TAKE requester to NO.

As we have seen, each image of an animation is normally saved to disk when it is finished. With save images set to NO, the animation will be compressed as it is rendered. Images will still be written to disk, but the image files will be deleted as soon as they are not needed. At most, five images will be retained at one time.

The drawback to this kind of animation is that any interruption will make it necessary to re-start the entire process from the beginning.

Color and exposure locking

There are two variables in a scene that can cause radical changes in the appearance of an image. These are the scene's exposure and the image's color map. Both of these items are determined automatically when you render an image with START, and are usually no cause for concern.

If each image were computed with an independent automatic exposure value and one object were to move closer to a lamp, then the overall brightness of the scene would be reduced to prevent 'burn out' of the object near the lamp. This would probably be a desirable action except for its effect on delta compression. If the overall image is darkened, most pixels would differ from one image to the next and the efficiency of delta compression would suffer.

When Sculpt Animate generates an image without using HAM mode, the image is restricted to a color map of 16 or 32 colors, depending upon the resolution used. Each color is chosen from a palette of 4096 possible colors in such a way as to best represent a particular scene. The process of delta compression requires that each frame uses the same set of 16 or 32 colors. This also applies to HAM images, because the 4096 colors that HAM can display must be derived from a base color map of only 16.

When the first frame of an animation is rendered, Sculpt Animate will determine the best color map and exposure for that image. These values will then be 'locked' throughout the animation.

You can have a little control over the color map that will be chosen for your animation. If RENDER FRAME is used to render the most representative frame of the animation, that frame will determine the color map and exposure. If you change your mind and want to start from scratch, click the DELETE ALL IMAGES button. This resets the locks so that the next frame rendered will determine their value.

If, for some reason, you want the exposure value to change during the course of the animation, you can 'animate' the observer's manual exposure setting, saving different values for different key frames in the animation. This will usually make the compressed RAM animation much larger, though. The manual exposure control can also be used to change the overall brightness of the animation.

ADVANCED TECHNIQUES

Animating the observer

Dramatic effects can be achieved by moving the observer's location and target during an animation. The visual effects produced this way can be very exciting, even if only SKETCH or PAINTING mode is used. Often, more information can be conveyed in this way than by using a static viewpoint. As with objects, the observer can be animated either by key frame animation or by global animation.

In global animation, the observer's location and target must be attached to paths in the global scene's hierarchy. The observer's location is called "LOCATION", and the target is called "TARGET". These are 'special' names in the hierarchy, and if they are entered into it they will be labeled accordingly.

When attached to a path, the location or target will follow that path precisely. The location and target should always be placed on separate paths, otherwise they would both be in the same place every frame! The target can be attached to the same path as an object, this will cause the observer's view to 'track' that object's local origin.

In key frame animation, the location and target may simply be placed differently in different key frame scenes. They will move smoothly from one position to the other when animated.

Other observer attributes can be animated in key frames, as well. Specifically, the lens, exposure, and tilt. Simply set these parameters differently in the key frame scenes, and they will change smoothly in the animation. As usual, key frame animation gives you the maximum creative freedom, at the cost of a little extra work.

If there are any key frame scenes defined for the animation, then the observer will be positioned according to those scenes. In this case, the observer's position in the global scene will be ignored, unless it is named in the global scene's hierarchy.

Even if a static view is being used, this is an important fact. If you want to change the observer's location for a large key frame animation, you can make the global scene's observer 'take precedence', simply by putting the names LOCATION and TARGET anywhere in the global scene's hierarchy (the highest level will do).

The only drawback to animating the observer is that the compression ratio for RAM animation is reduced, often dramatically, because most of the pixels that make up each image will change from frame to frame. This may be of less concern if you have a hard disk or other enormous storage medium.

When such drastic differences exist between frames, the benefit of normal RAM animation compression is reduced. In this case, you might as well click the RAM animation gadget to ECONOMY to speed things up.

When the compression efficiency is reduced, of course, the animation playback speed will suffer as well, since the processor will have that much more data to move between frames.

Later in this chapter, you will see how an animation's images may be recorded on a professional quality videotape recorder. This bypasses any need of RAM animation, and all its restrictions.

Animating lamps

The effect of moving lamps in a scene can be equally dramatic, although the cost in processing and compression can be even higher than that of moving the observer. Nonetheless, if the animation is to be finally rendered in photo mode, the moving shadows and highlights will give the scene an amazing illusion of solidity and reality.

In key frame animation, the lamp does not exist in the global scene, but is positioned separately in each key frame. You can also change the brightness and even the color of

ADVANCED TECHNIQUES

each lamp. If multiple lamps are used, then in-betweening problems can occur unless you drag the lamps with the Grabber rather than creating new ones for each scene.

In global animation, a lamp may be named and attached to a path, just as if it were an object. Use the command `EDIT NAME INDICATED LAMP` to do this.

Model Animation

In the movies, a classic animation technique is to animate models. A single picture is taken and then the model is moved a small amount and a second picture is taken. The process is repeated until a complete animation has been assembled.

This technique can employ many different types of models, frequently the models are rigid but with many jointed parts that can be set in particular positions. Alternatively, a malleable medium like clay can be used, which may require more skill but has almost unlimited creative potential.

In Sculpt Animate, solid models may be built and their hierarchical object structure used to help manipulate them.

We have already stressed some advantages of creating such a model as a collection of parts. For instance, a human model could comprise many elements constructed individually. A forearm could first be fashioned with the spin tool as a tapered cylinder with rounded ends. The expand command could be used to introduce just the right degree of asymmetry. Finally, some judicious manipulation of individual vertices could add realism.

An upper arm could be made in the same manner, and the two parts made to overlap in space, so they appear to be joined at the elbow. This is often quite adequate. Such rigidly 'joined' objects lack the flexibility of skin, however, so creases can appear when they are bent and repositioned.

This rigidity can be avoided by connecting the parts with faces. This requires a good deal of skill, and not a little bit of patience. The trick is to create 'fillets' of face-forming edges that link one part with the other. Any vertices within the joint that are not part of a visible face may then be erased. The vertices on either side of the fillet may be named differently, so when the joint moves the faces they form will stretch and contract to form a pliable 'skin'. With foresight, you can arrange things so that the UNSLICE command can be used to link the parts.

The complete human model could be named in a hierarchical manner, as suggested in Chapter 9. It then becomes easy to select either a single component or a whole group of them. By placing the cursor at the location of a joint and using the rotate gadgets, the model can be manipulated just like a jointed doll. Although some skill is required, the model can be moved to emulate any physical activity, such as walking.

The labor of animation is reduced by employing key frames. A walking man animation requires fairly frequent key frames, say one every four or five frames, but that is still much better than having to build every scene.

It is a good idea to place curves in the global scene, even if you are not performing global animation. These curves act as guides for positioning the model and help to ensure that the motion is smooth and fluid. Just as the eye can detect irregularities in a hand drawn line, it can also detect irregularities in motion.

Objects should not normally change their velocity instantaneously. Have velocity build up and decrease steadily. Remember that velocity is determined by the difference in position of a vertex from one frame to the next. The greater the difference, the greater the apparent velocity.

ADVANCED TECHNIQUES

Object Cycling

There is a special feature of the global scene's hierarchy, it is able to control the very existence of objects in different parts of the animation. This special feature takes place whenever an object's name begins with a numerical digit. Such names are analyzed as follows:

- 1) If the name starts off as a number followed by a minus sign followed by a second number, then the vertex is only loaded if the frame number lies in that range. A vertex with the name '17-24apple' will only be loaded into frames 17 to 24.
- 2) If the name starts with a number and is followed by a slash and a second number, then the vertex is only loaded for the frame indicated by the first number, and then at intervals indicated by the second number. A vertex with the name '3/5orange' would only be loaded for frames 3, 8, 13, 18 etc.
- 3) If the name starts with a number, but does not satisfy rules 1 and 2 above, then the vertex is only loaded for the frame with that number. The vertex named '17banana' is only loaded for frame 17.

These three rules let you perform something called object cycling. Suppose that you wish to animate an object that moves in a regular manner and repeats the motion many times. You can make a composite object that includes all possible configurations, and name them so that they are selected in turn for successive frames.

Suppose that the name Henry is applied to a human figure that has eight pairs of legs, each in a different position. The first pair of legs has the name Henry+1/8legs, the second pair Henry+2/8legs, etc. If this multi-legged Henry is placed in the global scene, only the first pair of legs will be loaded into the first frame, the second set will be loaded into the second frame, and so on.

You may have noticed that this particular Henry moves his legs back and forth, but he doesn't actually walk. To make Henry walk, he must be attached to a path. The speed along the path should be carefully chosen so that Henry's feet do not seem to skid.

Animated objects

Henry is an example of an animated object. He could be saved as an object and used for other animations. Just as you save interesting still objects for use in other scenes, you can also save moving ones. There are many other ways to build animated objects. Look at the scenes that are included on the disks distributed with Sculpt Animate. Some of them contain animated objects, study them to see how they work.

Animated objects can be made by attaching objects to paths, by object cycling, or by using both techniques simultaneously. Let us go through the steps for creating a spinning sphere. At first glance, the only way that we could build a spinning sphere is by object cycling, but there is a better and much easier way.

First build a sphere in the usual way, and then construct a hierarchy with a parent called 'spinning' and an offspring called 'sphere'. Select each vertex of the sphere and assign each vertex to the name 'sphere'. Set the local origin to the center of the sphere.

Now we must create a path that spins an object without moving it. Deselect the sphere and add a circle. Proceed as we did with the path named 'orbit' in Chapter 11. Remember to rotate the tumble axes around the path, as was explained in the 'Global vs. Key frame' section of that chapter. Name the path 'spinning'. If we stopped at this point, we would have a sphere spinning about an axis, but following a circular orbit. To avoid the orbiting motion, select the path and shrink it down with the EDIT DO EXPAND command. You could keep clicking on the 1/2 gadget until the path shrank to a single point, but it is better simply to make the circle much smaller than the sphere. That

ADVANCED TECHNIQUES

way you can still go in and easily modify the tumble axes if you need to.

Backgrounds and foregrounds

Although Sculpt Animate can be used to synthesize any scene, sometimes it may be desirable to combine renderings of three dimensional objects with background or foreground images produced by other programs, such as paint programs.

You can specify background and foreground images in the appropriately labeled boxes of the MODIFY TAKE requester, GLOBAL mode.

These particular gadgets require the complete AmigaDOS pathname. Sculpt animate will also assume that the file's name has the extension '.image', but do not type this extension into the gadget. For example, if your background image is named 'clouds.image' and its in the 'images' drawer of a disk named 'Stuff', then you will type in 'Stuff:images/clouds'.

As each image of the animation is rendered, the rendered image will be combined with the foreground and background images according to some simple rules:

- 1) Any pixel of a HAM foreground image that is not totally black will be shown in the final image.
- 2) Any pixel of a non-HAM foreground image that is not color zero will be shown in the final image.
- 3) If the world's sky is SOLID or GRADUATED, no background image pixels will be visible.
- 4) If the world's sky is NONE, the background image will replace any part of the sky area that is not blocked by a foreground image.

In other words, the rendered image will be 'sandwiched' between the foreground and background images, with the former showing in front.

Neither the foreground nor the background will have any effect on the rendered imagery itself, or be affected by shadows or scene lighting. Mirrored and glass objects will look as if the sky were the SOLID color that was set before the sky color gadget was toggled to NONE. The solid-sky color is remembered, even though it is inactive otherwise.

The foreground and background images may be any standard IFF files, including HAM images. Their image size and mode should not differ from the scene's settings. Sculpt animate will attempt to resolve any such differences, but it could significantly increase rendering time. Pixel dimensions for Sculpt Animate's various sizes and resolutions are listed in chapter 3.

If you must use a background image that differs in size or mode from your animation, you can avoid having it 'fitted' to your scene more than once. To do this, render a one-frame animation in the size and mode you want, using the desired image as a background for a totally empty global scene, with no sky or ground. You can use the resulting corrected image as your background in the full animation.

This suggests that background images can be generated entirely with Sculpt Animate. In this way, distant objects that do not move need only be rendered once. For complicated scenes, this could represent a substantial time saving when making a long animation.

The foreground image can also portray objects as though they were within the scene. Suppose all an animation's action takes place near the center of the screen, but which has complicated objects in the foreground or to either side of the action. As long as no shadows or reflections depend on these objects, you could render an image of them with no sky or ground, and use that image as a foreground for the full animation. Leaving these objects out of the full animation can reduce the total amount of rendering time.

ADVANCED TECHNIQUES

In the ray trace modes, some time savings could be realized by rendering one image of an animation, and 'blacking out' the area of motion with a HAM paint program. The foreground pixels will usually be faster to render than a full ray trace of the same pixel, even if all the objects are still in the scene. Of course, the restrictions to movement and shadows still apply.

Motion blurring

When an animation has rapidly moving objects, each image is sharply defined and the eye perceives the motion as jerky rather than smooth, even at a frame rate of 30 frames per second. A comparable film image would appear smooth because each individual frame is slightly blurred.

This blurring can be simulated in software by turning on the Motion blur switch prior to rendering a Take. Animations will take longer to produce, and because the blurring is achieved by blending together adjacent images, very rapidly moving objects will not have a correctly blurred image, so some jerkiness will remain.

Single frame controllers

The restrictions associated with RAM animations can be avoided if each frame is directly recorded on a video tape recorder, one frame at a time. Unfortunately, most consumer grade recorders are not able to cleanly record one frame at a time, so this technique is currently limited to the video professional.

If you plan to purchase a recorder and intend to use it for single frame recording, look for a recorder with 'flying erase heads', and make sure it can record single frames cleanly at the exact location you desire, without skipping or the image breaking up. Naturally, there must be some mechanism for controlling this.

Although some recorders can be operated manually to record single frames, it is preferable to use special hardware called a single frame controller, to act as an interface between the Amiga and the recorder. Sculpt Animate is designed to work with a special piece of software called a driver in order to communicate with the single frame controller.

A custom driver will have to be written for any particular controller hardware, but only the driver must be changed. Sculpt Animate itself can operate any controller device for which a driver has been written.

To use a single frame controller, you must set the RAM animation gadget to NO. Two new gadgets will appear in the Modify Take requester. One is called 'Frame controller' and the other is called 'Frame buffer'. If you click on one of these gadgets it will toggle to YES and a file requester will pop up and ask you to choose from among the appropriate drivers that are present in the system 'DEVS:' directory.

Generating an animation to a frame controller is almost the same as for RAM animation; click RENDER ALL to begin. The images will be sent to the controller in the same way they would be compressed, either as rendered or all at once afterwards, depending on the state of the 'save images' gadget.

Frame buffers

A frame buffer is a piece of hardware that generates an image independently of the Amiga's graphics hardware. Depending upon the design, it may generate a higher resolution image than the Amiga, or it may display more than 4096 colors, or both. Using a frame buffer can result in better quality images than could be produced by the Amiga's internal graphics hardware.

The frame buffer is accessed by means of a driver similar to the driver for a single frame controller. Its installation and use must be described by the driver's documentation.

ADVANCED TECHNIQUES

To select a frame buffer driver, click on the Frame buffer gadget to change it from NO to YES, and then select the driver using the file requester that will pop up.

Sculpt Animate will interrogate the driver to determine what size image is required, and other properties pertaining to the frame buffer. No other special action is required, although when rendering is started, the Amiga display may or may not show the images as they are generated.

If the 'save images' gadget is set to YES, the images will be saved to disk in a compressed format similar to IFF's ILBM. The format used is not IFF compatible, however, because it will usually exceed ILBM's maximum resolution.

Connecting up the hardware

As with any other hardware, carefully follow the directions that are supplied. A typical setup has an Amiga, a professional videotape recorder (VTR), a single frame controller, and a frame buffer.

The VTR will have some form of control hook-up connected to the frame controller, which in turn will be connected to the Amiga, probably through the serial or parallel port. This is the control link.

The VTR must also have video input, this can come directly from the Amiga's composite output, from an NTSC encoder or GenLock connected to the Amiga's RGB output, or from a frame buffer connected to the Amiga's expansion bus.

In a typical animation recording session, the program will display each image, either on the Amiga's display or by passing it to a frame buffer driver, and then signal the frame controller that the image is ready. This is repeated until every frame is recorded.

There is a vast amount of video hardware on the market right now, and very little of it is truly suitable for single

ADVANCED TECHNIQUES

frame recording. Before acquiring hardware for this purpose, be sure to study all of the latest literature and periodicals, and consult both your local Amiga and video dealerships. Unfortunately, we cannot even hope to adequately introduce the art of videography in this manual.

Sculpt Animate drivers have been written for a few video devices, and more are in the works. If you have a particular setup in mind, or you have built your own video hardware, it is possible to write your own software drivers, although considerable programming experience is required. Sourcecode for existing frame buffer and frame controller drivers can be purchased from Byte By Byte Corporation.

Using a GenLock

A GenLock has other uses in Sculpt Animate, besides encoding RGB to NTSC for recording to videotape. A GenLock's main purpose, after all, is to overlay computer graphics onto an external video source.

The GenLock works according to a relatively simple principle (although the details are technically complex). The GenLock intercepts the Amiga's RGB image output and encodes it. As each pixel is transmitted, the Amiga includes a signal which indicates whether the pixel's color is 'color zero'. When its genlocking feature is activated, the GenLock will replace any of the Amiga's color zero pixels with the corresponding pixel of an external video source, such as a VTR or video camera image. All other Amiga pixels are transmitted normally.

In order to permit the use of Sculpt Animate animations with GenLocks, the program ensures that the black color used for the 'Sky color NONE' background is color zero in the Amiga's color map. When a GenLock is being used, the sky area will be 'transparent' in any animation or image that is rendered with this setting, so that an external video source may be seen behind the animation's action.

ADVANCED TECHNIQUES

As with background-foreground images, the GenLocked image will not be reflected or transmitted by mirrored or glass objects. You can simulate such reflection to a certain extent, though, by setting a SOLID color for the sky that you want to be reflected. Of course, the genlocked image will not be visible unless you change the sky color to NONE, but the color you set for SOLID will be remembered, and used for reflections and glass transmissions.

GenLocking the Tri-View

Sculpt Animate automatically supports the use of a GenLock to overlay the Tri-View editing environment onto an NTSC video source. If a GenLock is connected to the system, the Tri-View's background will be color zero (the 'transparent' color). This facility can be useful for digitizing shapes, but is naturally limited to the screen's resolution.

There may be times when you want the normal Tri-View screen to appear, even if a GenLock is attached to the system (we encountered this problem when making a tutorial video). A program entitled 'GenLockToggler' is present in the 'Misc' subdirectory of your '68020/68881 VERSION' distribution disk. You can run this program to 'hide' your GenLock *before* you run Sculpt Animate. Run the GenLockToggler again to reveal the GenLock afterwards.

Be a little cautious with the GenLockToggler program. It can be used to 'reveal' a GenLock whether there is one or not. This is probably not what you want to do.

1. Introduction

2. Methodology

3. Results

4. Discussion

5. Conclusion

6. References

7. Appendix

8. Acknowledgments

9. Contact Information

10. Declaration of Interest

11. Funding Source

12. Author Contributions

13. Data Availability

14. Ethics Approval

15. Conflicts of Interest

16. Supplementary Materials

17. Correspondence

18. Received Date

19. Accepted Date

20. Published Date

21. Copyright

22. License

CHAPTER 13

BY THE NUMBERS

So far, this manual has presented the techniques of building, manipulating, and animating 3D models as an artistic pursuit. The aim has been to provide an environment for freehand model building, so that no mathematical skill is required beyond some ability to visualize things in three dimensions.

For many people, however, the scale and accuracy of the models they build is important. This chapter presents a number of tools that can be used to precisely scale and position the objects in your scenes. To understand this facility, you will need to learn a little three-dimensional geometry.

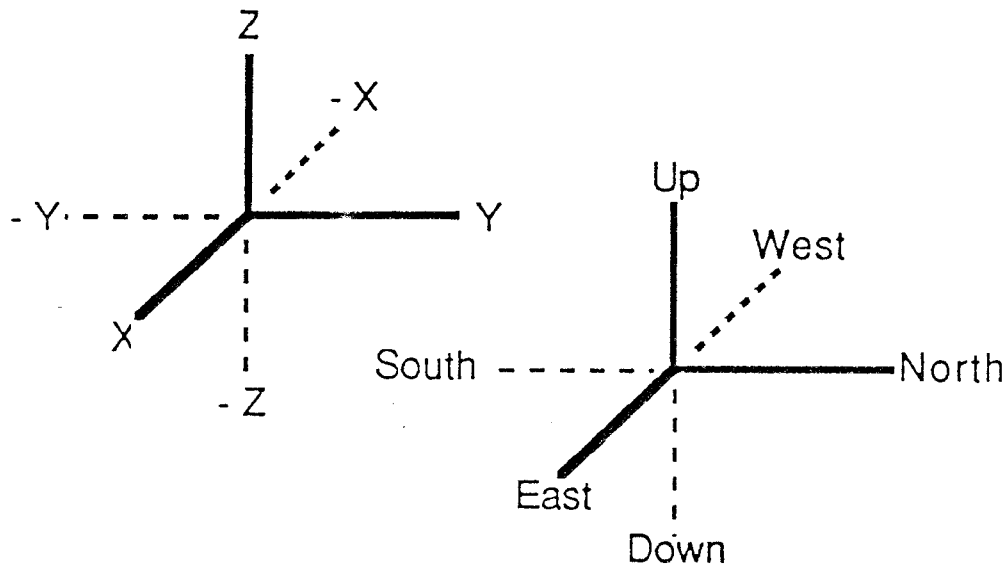
Sculpt Animate coordinates

You can think of the Sculpt Animate universe as an enormous cube of space. When you run the program, the Tri-View represents a much smaller cube within this space, situated in its exact center. As you use the Tri-View, you move it around within the universe, expanding and shrinking your view so that it contains more or less of the available space.

An essential part of Sculpt Animate is that every vertex, lamp, or other object has a precise location within the universe. Any location in a 3D space can be described with only three numbers, each of which describes an offset in one of the three dimensions. This sort of arrangement is called a Cartesian coordinate system (named after René Descartes). The street-locator grid on a fold-up city map is a cartesian system in two dimensions. A 3D system is similar, except that it can also 'map' vertical offsets.

BY THE NUMBERS

Any given location in Sculpt Animate is described in terms of the cardinal directions East, North, and Up. These directions correspond to the 'X', 'Y', and 'Z' of a traditional coordinate system in mathematics.



Sculpt Animate Coordinate System

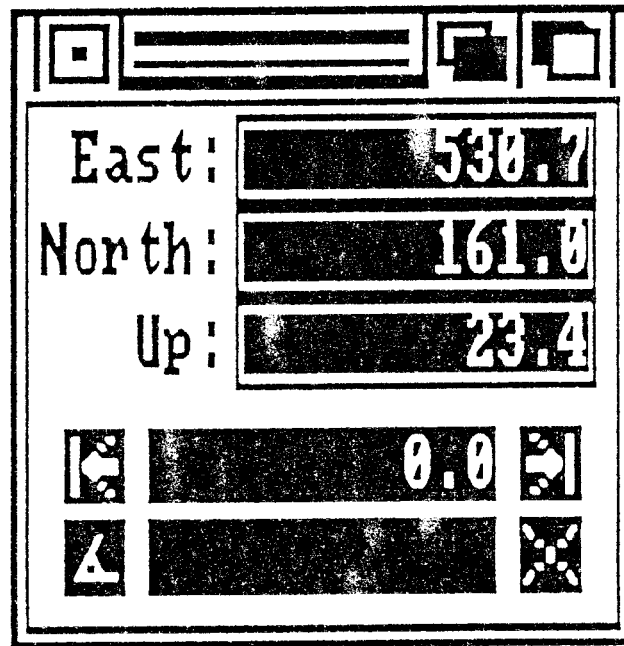
The Coordinates window

If you execute the command EDIT COORDINATES, a new window called the coordinate window will pop up. This window has a drag bar, so you can move it to a location where it will not obstruct the Tri-View.

When you invoke this window, a requester will inquire how many decimal places you want to use. Sculpt Animate's universe is divided into a fixed number of indivisible units, but most people will want to work with a divisible unit of measurement. By choosing a particular number of decimal places, you determine what fraction of your 'working' unit each Sculpt Animate unit will be. For instance, if you choose one decimal place, the smallest fraction you will be able to work with is 0.1, or one tenth of your working unit.

BY THE NUMBERS

Many people wonder why their working unit of measure isn't marked as inches, or feet, or meters, etc. It is left up to the user to decide what one 'unit' in the coordinates window will be equivalent to, be it parsec, cubit, or angstrom. All Sculpt Animate must know is the largest number of decimal places you'll need to express a fraction of one working unit.



The Coordinates Window

Once you have entered the number of decimal places, a new window will appear on the screen. This window displays the cursor position relative to an origin. The position is represented as a distance East, a distance North and a distance in the Up direction, relative to the coordinate origin. If the displayed value is negative, it represents an offset in the opposite direction.

At this point, you may drag the coordinates window to one side, and continue working in the Tri-View as usual. The readout of the cursor's position will be updated as you move it around.

When you are working with the coordinates window on the screen, you can also click on one of the coordinate boxes and edit or enter a number. When you press the return key,

BY THE NUMBERS

the Tri-View cursor will jump to the location that you specify.

The coordinate origin

The center of a coordinate system is called its 'origin'. Coordinate values are read as offsets from this location.

The origin of the coordinate window starts out at the absolute origin, in the center of the Sculpt universe. It is often convenient to work with coordinates based on some other place in your scene. To do this, you can move the coordinate origin by placing the cursor at the desired location and clicking on the X shaped gadget in the extreme lower right of the coordinates window.

Even though the coordinate origin can move about, you can find the absolute origin at any time by holding down the right Amiga key and clicking the Center Tri-View gadget. This places the cursor back at the absolute origin.

If you are building an object from a set of plans, it is a good idea to set the origin at a convenient location. For example place the origin at one corner of a house and then the numbers in the coordinates window can directly correspond to measurements on the plans. If you are building a complicated object from many simpler ones, you can reset the origin for each object.

Once an object is built, it can be very simple to position it precisely. Usually there will be a vertex of the object that you will want to be precisely aligned in a larger context. You can use the command `EDIT SNAP CURSOR TO VERTEX` to place the cursor precisely on this vertex, then turn on the GRABBER and type a new location for the vertex directly into the coordinate window.

If you want to place one object relative to another object's center, `DESELECT ALL`, select the reference object and do a `SNAP CURSOR TO CENTER`, then zero the coordinates window. Now `DESELECT ALL` and select the

object you want to move. Use a SNAP CURSOR command to place the cursor precisely, relative to this object. Finally, turn on the GRABBER and enter zeros into all three coordinates, putting the cursor in the center of the reference object.

If you want an object placed precisely on the ground, you can select the object, indicate its lowest vertex, do a SNAP CURSOR TO VERTEX, and turn on the Grabber tool. Now simply zeroing out the 'Up' coordinate will move the object so that its bottom vertex is precisely at ground level. This works if the coordinates are still zeroed at the 'absolute' origin. If not, click Center Tri-View while holding the right Amiga key to find the ground level.

These are just a couple of examples of how the coordinates window can be combined with the SNAP CURSOR commands to precisely place objects in the Tri-View.

The tape measure

The fourth number in the coordinates window is the readout for an automatic tape measure.

There are also two end marker gadgets, one to either side of this readout. Clicking one of these gadgets will place one of a pair of marker icons at the current cursor position. These two markers represent the ends of the tape measure, as soon as both are placed in the Tri-View, you will see the distance between them displayed in the readout.

If a vertex is indicated when you click a marker gadget (i.e., the cursor is close to it in the selected window), then that marker will be 'attached' to the indicated vertex. If the vertex is moved in the Tri-View, the marker will move with it and the tape measure readout will be updated accordingly.

A marker icon will be purple if it is attached to a vertex. Otherwise it will be blue.

BY THE NUMBERS

You can use the tape measure to scale objects precisely. Say you want a sphere with a specific radius. First, DESELECT ALL and add a sphere in the normal way. Select the sphere and snap the cursor to its center. Click Shrink Tri-View until no vertex is visible under the cursor, then click one of the marker gadgets. Expand the Tri-View again, indicate any vertex, and click the other marker gadget.

Now that the tape measure is set, call up EDIT DO EXPAND. Each time you adjust the sphere's size, the tape measure will reflect the new radius. EXPAND's lower slider lets you scale sizes with an accuracy of almost one part in 10,000.

When you are using spheres, it might be convenient to place a vertex at the center of each sphere, and connect it to the rest of the sphere with an edge. This lets you use SNAP CURSOR TO VERTEX, instead of having to deselect everything, select the sphere, and snap to its center. It also provides a convenient anchor vertex for the tape measure and local origin.

Of course, if all you want to do is scale the sphere, you could mark diametrically opposite vertices and read the measure as the sphere's diameter instead of radius. This entirely avoids the need to use SNAP CURSOR TO CENTER, but it wouldn't show off as many techniques!

The protractor

Just as you can use the tape measure readout for measuring distances, there is also a way to measure angles. This is done with the readout on the bottom of the coordinates window. Notice the 'angle' gadget to the left of this readout (opposite the origin gadget). Clicking this gadget places a similarly shaped icon at the cursor's current location.

This icon is the protractor marker. It works just like the two tape measure end markers, in that it may be placed independently or attached to an indicated vertex. The

protractor works with the tape measure icons to mark an angle to be measured.

To see how this works, construct a triangle in the Tri-View. Move the cursor to any of the triangle's vertices, and click on the left hand tape measure gadget, then move the cursor to another vertex and click on the right hand tape measure gadget. Finally move the cursor to the third vertex and click on the angle icon at the lower left of the coordinate window. All the markers should be displayed in purple, indicating that the marker is locked to the vertex.

The bottom box in the coordinates window will now display an angle measurement. This measurement is the size, in degrees, of the apparent arc between the two tape measure markers, as measured from the protractor marker. In this case, this is the angle at the triangle corner which is marked by the protractor. The value's precision is two decimal places, regardless of the precision of the other numbers in the coordinate window.

Deselect all but the one vertex and use the Grabber tool to move the vertex around. Watch how the angle displayed in the coordinate window changes. You can see that this tool can be used not only to measure, but also to construct objects by their angular measurements.

We have been using the protractor to measure angles in the plane of one Tri-View window, but by placing the various markers in three dimensions you can measure any angle, no matter what plane it lies in. This can be helpful when constructing models for scientific purposes.

The grid

If you are modeling objects that are built according to regular, integral measurements, a reference grid can be a great help. Sculpt Animate provides a three dimensional grid, a sort of 3D graph paper, that you can use to help build and position your objects.

BY THE NUMBERS

You turn on Sculpt Animate's grid with the command `EDIT GRID ON`. When you use this command, a requester will pop up asking for the grid size. This is the size of each grid cube in Sculpt Animate's 'fundamental' units, the least significant digit of the coordinate system.

The grid size can be made the same as your working unit by typing in the correct scaling factor. Thus, if your working unit is divisible by thousandths (0.001, three decimal places), then an equivalent grid size would be 1000 fundamental units.

The grid size can also be some multiple or fraction of your working unit. For instance, if you have decided your working unit is inches, with three decimals of accuracy, then you can specify grid lines at one foot intervals by entering a grid size of 12 times 1000, or 12000. A grid size of 125 would be equivalent to 1/8th inch grids.

The grid is displayed in the Tri-View as faint grid lines in each of the three windows. If the Tri-View is expanded to the point that the grid lines would be spaced too closely, then the grid will disappear until the Tri-View is shrunk again.

The grid is not just used as an aid to visual placement of objects. The commands `EDIT SNAP CURSOR TO GRID` and `EDIT SNAP SELECTED VERTICES TO GRID` both work as you might expect, moving the items in question to the nearest grid intersection.

`SNAP CURSOR TO GRID` can be used like the other `SNAP CURSOR` commands as an aid to positioning objects. Note that using the menus for a `SNAP CURSOR` command will not drop the grabber, as it usually would.

The grid can be removed with the command `EDIT GRID OFF`. If the grid is already on and you want to change its scaling factor, just call `EDIT GRID ON` a second time and enter a new size value.

CHAPTER 14

KEYSTROKES, FONTS, AND SCRIPTS

On the surface, the design of Sculpt Animate is such as to require the minimum of rote learning. Once you have grasped a few underlying concepts the rest should follow naturally. Likewise, if you do not use the program for a while, it should not take you long to get back up to speed.

The drawback of this approach, with its extensive use of menus, is that the program cannot be operated as quickly as a keyboard based program could, in the hands of an experienced user. It is quicker to press a key on the keyboard than it is to make a menu selection, provided that you are able to remember which key to press to produce the effect of a specific menu command.

To address this need, Sculpt Animate makes use of the keyboard in a number of ways. There are menu-defined keyboard equivalents, 'on-the-fly' user definable hot keys, keyboard-automated text font loading, and a scripting facility that allows Sculpt Animate to be 'programmed' to perform automated routines. Script routine 'macros' can even be attached to hot keys.

These are all rather advanced features, so much so that it's unlikely that any one user will make full use of all of them. Each of them addresses a specific need, however, and they are here if you need them.

Keyboard menu equivalents

As with most Amiga programs, there are a series of keyboard equivalents associated with the menus. When you scan through the menus you will notice that some commands have a notation like F1 to the right. This means that the

KEYSTROKES

command can also be executed by means of a keystroke, in this case the 'function key' marked 'F1'.

Other commands are marked with the symbol **4**, followed by a letter. This means that the command can be accessed by holding down the right Amiga key while pressing the indicated letter key.

The keyboard equivalent for a menu command will have precisely the same effect as the menu command itself would. Since the menu commands are all right there in the menus, it is easy to refresh your memory if you forget one.

There is one 'universal' keyboard equivalent. Pressing the escape key (marked ESC) will cause the last menu command to be repeated. This is particularly useful if you need to perform the same command many times in a row.

For example, if you need to erase a dozen edges, point the cursor to the first edge and execute the command EDIT ERASE INDICATED EDGE. Then point the cursor at the second edge and press the escape key. Continue this sequence and you will find that it is much quicker than using the menu.

User settable hot keys

The trouble with using the escape key as a short cut for a menu command is that it can only represent one command at a time. There is a way that each of the alphabetic keys, A through Z, can be assigned to a menu command, so you can set up as many as 26 single-stroke commands.

If you press one of these while holding down the ALT key, then the alphabetic key becomes equivalent to the last menu command that was executed. This assignment will remain in effect until you quit the program.

The problem with hot keys is that they require you to remember what is defined. Try to choose a letter that is reminiscent of the command you are defining, such as 'C'

KEYSTROKES

for SELECT CONNECTED VERTICES. The relationship needn't be alphabetic, either, for instance, 'X' could be a hot key for SNAP CURSOR TO CENTER.

There are fewer keys than commands, but you will find that certain commands are used together while you are doing certain types of editing. Because the keys can be freely redefined, you may find it useful to change your definitions as a project proceeds from one stage to another. This is a 'power user' technique, though, and not recommended unless you feel comfortable with it.

Fonts

There are still more ways in which the keyboard can be used to save time. 3D Font object library disks designed for Sculpt Animate can be purchased separately, but it can be difficult to place these font character objects in the scene by hand.

To ease the burden, Sculpt Animate has a keyboard-driven automated font loading mechanism which can be used with font disks laid out in a particular fashion.

For Sculpt Animate's purposes, a font is a set of scene files containing three dimensional models of solid characters. Individual fonts will differ in which characters they contain, but most will contain the upper case characters, and some will contain other characters as well.

To activate the type-in font feature, execute the command PROJECT LOAD FONT. A standard file requester will pop up to allow you to choose the directory in which the font's scene files reside. This time it is not necessary to select a particular file, just specify the correct volume and drawer, then click OK.

Once a font is loaded, pressing any key that has a graphic symbol will cause the program to search the font directory. If the character is found, it will be loaded into the

KEYSTROKES

current window at the position of the cursor, and the cursor will be advanced to the right by the size of the character.

The first character you type will be scaled so that its height is the same as the height of the current window. Subsequent characters will be scaled by the same proportion until the next time you execute LOAD FONT.

So long as a font is loaded, all the keys, whether or not their corresponding font objects are actually available, will be assumed to be connected to the font. User definable hot keys will not operate, nor will the script based keystroke macros that will be defined in the next section. To re-enable the other uses of the keyboard, you must execute the command PROJECT UNLOAD FONT

We mentioned that fonts must be laid out in a particular fashion. For one thing, it will be assumed that characters were originally constructed in the North view, although they can be loaded into any view. The characters should also be saved with Save Object, with the cursor placed at each character's baseline and some distance West of its leftmost vertex.

The cursor's left offset will be used to space each character from the previous one. The cursor's position in the other two dimensions will be used to align the characters in a row as they are loaded.. By carefully placing the cursor when each character is saved, both proportional spacing and correct handling of descenders can be provided for. Free vertices may also be saved with the character object, if necessary, to define the space it must occupy. These vertices will be ignored when the character is loaded.

If, upon saving, the cursor was not placed to the left of the character's leftmost vertex, Sculpt Animate will try to estimate an appropriate spacing. The cursor will determine the character's baseline and depth alignment in any case.

It is assumed that all a font's character scene files will be stored in a single directory, and that scenes for different fonts will be placed in separate directories.

KEYSTROKES

Also, each character scene must be named correctly in order for Sculpt Animate to find it. The program will use these rules when looking for a character's scene:

- 1) If a lowercase alphabetic character is pressed--'a', for example--then Sculpt Animate will look in the selected font directory for files named like these two, in this order: 'a.scene', and 'a_3D.scene'.
- 2) If an uppercase alphabetic character is pressed--'A', for example--then Sculpt Animate will look for files named like 'ua.scene' and 'acap_3D.scene'.
- 3) If a numeric key is pressed--'9' for example--then Sculpt Animate will look for '9.scene' and '9_3D.scene'.
- 4) If a slash (/) or colon (:) is pressed, Sculpt Animate will look for 'slash.scene' or 'colon.scene', respectively.
- 5) If any other printable character is pressed, including the ALT modified characters listed in the AmigaDOS manual, Sculpt Animate will first append '.scene' to that character, then '_3D.scene', and try to open files with these names. Examples: '&.scene', '&_3D.scene'.
- 6) If the character scene file still hasn't been found after the preceding rules have been applied, the '.scene' suffix will be affixed to the two-character hexadecimal equivalent of the desired character's ASCII code (for example, '3F.scene' if you pressed '?'), and one last attempt will be made to open the file.

The last rule provides for characters which can be awkward to use in AmigaDOS file names. Some of these are: asterisk (*, hex equivalent 2A), space (,20), equals (=,3D), plus (+,2B), single quote (' ,27), double quote (" ,22), question mark (?,3F), percent (% ,25), sharp (# ,23),

KEYSTROKES

vertical bar (|,7C), semicolon (;,3B), and parentheses (28 and 29).

Scripts

Although most people will want to operate Sculpt Animate in an interactive mode, it can also be controlled from a script command file. Script files are definitely an advanced tool, and will probably be of most interest to those with some knowledge of computer programming.

A script file is a plain text file that has been typed in and saved with a text editor, such as the Amiga CLI's 'ED' command. All script file names must end with the extension '.script'.

Once you have created and saved your script file, you can start its execution with the command PROJECT LOAD SCRIPT. This will invoke a standard file requester, which you can use to pick which script to load.

When you click OK, Sculpt Animate will go through the script interpreting the commands in sequence. For instance, the following script will load in a scene, render it, and save the resulting image.

```
erase all  
load scene "test.scene"  
start  
save image "test.image"
```

A special exception to this behavior is initiated by the MACRO command. This command defines a hot key 'macro', a set of script commands that can be executed by a single keystroke. The MACRO command is explicitly defined at the end of the next section. For an example, the following script defines a hot key that will grab a connected object by the indicated vertex and align it to the grid.

```
macro z
    snap cursor to vertex
    select connected
    grabber on
    snap cursor to grid
    grabber off
    deselect connected
mend
```

The command MEND ends the macro definition. The defined sequence of commands is indented here, but this is not necessary. It is, however, a recommended technique, since it makes the script much easier to read.

If a file called 'startup.script' exists in the current directory when Sculpt Animate is run, then this script file is automatically executed. The startup script can be especially helpful for defining macros that you will want to use all the time. Note that you can define hot keys in the startup script as macros, as in the following example.

```
macro t target      mend
macro l location    mend
macro g grabber     mend
```

Notice that several statements may be placed on the same line. This should be reserved for fairly simple macros, though, to keep things readable.

As the ultimate step in using scripts, people with programming skills can write script-generating programs to create very sophisticated, special purpose scenes. A typical application of this technique would be to create specialized objects that lend themselves to mathematical techniques.

Script writing programs would make extensive use of the cursor placement and edge creation commands described in the next section. These commands operate on the principle that (East, North, Up) would be equivalent to the coordinates (X, Y, Z) in traditional mathematics. Some

KEYSTROKES

example script-generating programs may be found on your Sculpt Animate distribution disks.

Script commands

The following script commands are directly equivalent to menu commands. When part of the command is shown in lower case, that part is optional and may be abbreviated. It is a good idea to place each script command on a separate line.

Script command

LOAD CODE
LOAD WORKbench
SHOW IMAGE
SHOW ANIMATION
SHOW PREview
UNLoad IMAGE
UNLoad ANIMATION
UNLoad PREview
UNLoad FONT
UNLoad CODE
UNLoad WORKbench
SElect ALL
SElect CONNected
SElect INDicated VERtex

SElect SWAp
SElect INDicated EDGE
SElect INDicated SPLine
SElect INDicated PATH
DESelect ALL
DESelect CONNected
DESelect INDicated VERtex

DESelect SWAp
DESelect INDicated EDGE

ERASE SElected VERTices
ERASE SElected EDGes
ERASE INDicated VERtex
ERASE INDicated EDGE
ERASE INDicated LAMP
ERASE ALL LAMPs

Menu command

PROJECT LOAD CODE
PROJECT LOAD WORKBENCH
PROJECT SHOW IMAGE
PROJECT SHOW ANIMATION
PROJECT SHOW PREVIEW
PROJECT UNLOAD IMAGE
PROJECT UNLOAD ANIMATION
PROJECT UNLOAD PREVIEW
PROJECT UNLOAD FONT
PROJECT UNLOAD CODE
PROJECT UNLOAD WORKBENCH
EDIT SELECT ALL
EDIT SELECT CONNECTED
EDIT SELECT INDICATED
 VERTEX
EDIT SELECT SWAP
EDIT SELECT INDICATED EDGE
EDIT SELECT INDICATED SPLINE
EDIT SELECT INDICATED PATH
EDIT DESELECT ALL
EDIT DESELECT CONNECTED
EDIT DESELECT INDICATED
 VERTEX
EDIT DESELECTS WAP
EDIT DESELECT INDICATED
 EDGE
EDIT ERASE SELECTED
 VERTICES
EDIT ERASE SELECTED EDGES
EDIT ERASE INDICATED VERTEX
EDIT ERASE INDICATED EDGE
EDIT ERASE INDICATED LAMP
EDIT ERASE ALL LAMPS

KEYSTROKES

ERASE INDICATED PATH
 ERASE INDICATED SPLINE
 ERASE ALL
 SUBDIVIDE
 FILL
 UNSLICE
 REFLECT
 MAKE SPLINE
 MAKE PATH
 SHOW PATH
 HIDE

REVEAL

MAKE TV BIG
 MAKE TV SMALL
 ADD DUPLICATE
 ADD LAMP
 ADD VERTEX
 ADD EDGES
 SNAP CURSOR TO VERTEX
 SNAP CURSOR TO CENTER
 SNAP CURSOR TO CENTROID

SNAP CURSOR TO GRID
 SNAP SELECTED VERTICES TO GRID

SNAP CONNECTED VERTICES TO SPHERE

SNAP SELECTED VERTICES TO PLANE

GRID OFF
 EXTRUDE
 GRABBER ON

GRABBER OFF

WIREFRAME
 SKETCH
 PAINTING
 SNAPSHOT
 PHOTO
 SCANLINE PAINTING

SCANLINE SNAPSHOT

LORES
 HIRES
 NO INTERLACE

INTERLACE

EDIT ERASE INDICATED PATH
 EDIT ERASE INDICATED SPLINE
 EDIT ERASE ALL
 EDIT DO SUBDIVIDE
 EDIT DO FILL
 EDIT DO UNSLICE
 EDIT DO REFLECT
 EDIT DO MAKE SPLINE
 EDIT DO MAKE PATH
 EDIT DO SHOW PATH POSITION
 EDIT DO HIDE SELECTED

VERTICES

EDIT DO REVEAL HIDDEN
 VERTICES

EDIT DO MAKE TRI-VIEW BIG
 EDIT DO MAKE TRI-VIEW SMALL
 EDIT ADD DUPLICATE
 EDIT ADD LAMP
 EDIT ADD VERTEX
 EDIT ADD EDGES
 EDIT SNAP CURSOR TO VERTEX
 EDIT SNAP CURSOR TO CENTER
 EDIT SNAP CURSOR TO
 CENTROID

EDIT SNAP CURSOR TO GRID
 EDIT SNAP SELECTED VERTICES
 TO GRID

EDIT SNAP CONNECTED
 VERTICES TO SPHERE

EDIT SNAP SELECTED VERTICES
 TO PLANE

EDIT GRID OFF
 TOOLS EXTRUDE
 TOOLS GRABBER

(when grabber is off)

TOOLS GRABBER

(when grabber is on)

OBSERVER MODE WIREFRAME
 OBSERVER MODE SKETCH
 OBSERVER MODE PAINTING
 OBSERVER MODE SNAPSHOT
 OBSERVER MODE PHOTO
 OBSERVER MODE SCANLINE
 PAINTING

OBSERVER MODE SCANLINE
 SNAPSHOT

OBSERVER MODE LO-RES
 OBSERVER MODE HI-RES
 OBSERVER MODE NO-

INTERLACE

OBSERVER MODE INTERLACE

KEYSTROKES

LOCation	OBSERVER LOCATION
TARget	OBSERVER TARGET
LENs NORmal	OBSERVER LENS NORMAL
LENs TELephoto	OBSERVER LENS TELEPHOTO
LENs WIDeangle	OBSERVER LENS WIDEANGLE
EXPosure AUTo	OBSERVER EXPOSURE AUTO
IMAge size TINy	OBSERVER IMAGE SIZE TINY
IMAge size SMAlL	OBSERVER IMAGE SIZE SMALL
IMAge size MEDium	OBSERVER IMAGE SIZE MEDIUM
IMAge size FULl	OBSERVER IMAGE SIZE FULL
IMAge size JUMbo	OBSERVER IMAGE SIZE JUMBO
IMAge size VIDEo	OBSERVER IMAGE SIZE VIDEO
ANTi aliasing NONE	OBSERVER ANTI-ALIASING NONE
ANTi aliasing GOOD	OBSERVER ANTI-ALIASING GOOD
ANTi aliasing BESt	OBSERVER ANTI-ALIASING BEST
STArt	OBSERVER START

The next group of script commands correspond to menu commands, but additional data must be supplied, usually that which is given to a requester. Items shown in angle brackets stands for this data as follows.

<filename> A complete file name, including the extension, enclosed in quotes. For example "littlehouse.scene".

<name> A hierarchy name, enclosed in quotes. Parents and offspring are separated by plus signs. If the hierarchy does not exist, it is created. For example "bill+leftarm"

<rgb4> Three numbers in the range 0 to 15 to represent red green and blue values. For example 15 15 15 would represent white.

<rgb> Three numbers in the range 0.0 to 1.0 to represent red green and

KEYSTROKES

blue values. For example 1.0
1.0 1.0 would represent white.

<number>

A numeric value.

<texture>

One of the following:

DULl
SHIny
MIRror
LUMinous
GLAss
METal

<side>

One of the following:

INWard
OUTward

<slope>

One of the following:

CALculated
SPECified

<knot-type>

One of the following:

REGular
CUSp

<direction>

Three numbers representing a direction. The first number is an eastward offset, the second northward and the third upward.

<character>

any keyboard character that produced something on the screen except white space when it is typed. For example an alphabetic character, a digit or a special character. Not a space, tab, return, ALT etc.

<script commands>

Any sequence of script commands except MACRO and

KEYSTROKES

MEND. The sequence can span more than one line.

When a reference is made to a coordinate, the current scaling as specified by the most recent COOrdinates command is employed.

When part of a command is optional, it is shown in square brackets. For example the command for loading a scene has the syntax

```
LOAD SCENE <filename> [LAMps] [OBServer]
[WORLD] [OBJects] [SELECTED]
[DESELECTED]
```

The actual command might be

```
load sce "littlehouse.scene" lamps obj selected
```

and mean that the littlehouse file is to be loaded, but only the objects and the lamps, with all loaded vertices being selected. The meaning of the following commands closely follows the menu commands. where it is not obvious, it is explained.

```
LOAD SCENE <filename> [LAMps] [OBServer] [WORLD]
[OBJects] [SELECTED] [DESELECTED]
```

```
LOAD IMAGE <filename>
```

```
LOAD OBJECT <filename> [SELECTED] [DESELECTED]
```

```
LOAD NAMED object <name> <filename>
```

```
LOAD TAKE <filename>
```

```
LOAD ANIMATION <filename>
```

```
LOAD FONT <filename>
```

KEYSTROKES

SAVe SCENE <filename> [LAMps] [OBServer] [WORld]
[OBJects] [ALL] [SELECTED]

SAVe IMAGE <filename>

SAVe OBJECT <filename>

SAVe NAMED object <name> <filename>

SElect NAMED vertices <name>

DESelect NAMED vertices <name>

ERase NAMED vertices <name>

MODify WIRE-frame colors <rgb4><rgb4>

EXPAnd <number> [NS] [EW] [UD]

Note that the expand command will expand selected vertices by the factor given by <number>. If none of the keywords NS, EW or UD are supplied, then the expansion is in all directions. NS stands for expansion in the north-south direction, EW means east-west and UD means up-down.

SPIn <number> <number> The first number is the number of spin steps and the second number is the spin angle. For example spin 12 360 would generate a 12 step full circle object.

MAKe HELix <number> <number>

The first number is the number of spin steps per turn and the second number is the total number of spin steps. For example make helix 12 48 would generate a four turn helix with 12 steps per turn.

ADD SPHere <number> <number>

The first number is the radius of the sphere and the second number is the number of subdivisions.

KEYSTROKES

ADD HEMisphere <number> <number>

The first number is the radius of the hemisphere and the second number is the number of subdivisions.

ADD CUBe <number>

The number is the size of the cube.

ADD PRISM <number>

The number is the size of the prism.

ADD DISK <number> <number>

The first number is the radius of the disk and the second number is the number of subdivisions.

ADD CIRcle <number> <number>

The first number is the radius of the circle and the second number is the number of subdivisions.

ADD CYLinder <number> <number>

The first number is the radius of the cylinder and the second number is the number of subdivisions. The height of the cylinder is twice the radius.

ADD TUBE <number> <number>

The first number is the radius of the tube and the second number is the number of subdivisions. The height of the tube is twice the radius.

ADD CONE <number> <number>

The first number is the radius of the cone and the second number is the number of subdivisions. The height of the cone is twice the radius.

NAME SElected VERtices <name>

NAME CONnected VERtices <name>

NAME INDicated PATH <name>

NAME INDicated LAMP <name>

KEYSTROKES

GRId ON <number>

The number specifies the grid spacing.

COORDINATES <number>

The number specifies the number of decimal places.
This command should normally be issued before any other script command that employs coordinate values or measurements.

MAGnet ATTract <number>

The number represents the strength of the magnet in terms of its standard value.

MAGnet REPel <number>

The number represents the strength of the magnet in terms of its standard value.

BITplanes <number>

BITPLANES 24 <filename> <filename> <filename>
 <number> <number>
 <number> <number>

The three filenames are for red, green and blue data.
The first number is the relative pixel width. The second number is the relative pixel height. The third number is the image width. The fourth number is the image height. All the number must be integers.

LENs SPEcial <number>

The number is the focal length of the lens in millimeters.

EXPOsure MANual <number>

The number is the exposure override value expressed as a percentage of the standard value. For example exp manual 50 would make the picture half as bright as normal.

DIThering <number>

The number represents the percentage of standard dithering.

KEYSTROKES

TILt <number>

The following commands have no menu equivalents, but correspond to requester functions.

ERASE TAKE

Remove a Take from memory.

FACe COLor <rgb>

SET FACe COLor

SMOothing OFF

SMOothing ON

SET SMOothing

TEXture <texture>

SET TEXture

BLEnd FACe COLor

FETch FACe SMOothing

FETch FACe TEXture

LAMp COLor <rgb>

LAMp BRighTness <number>

BLEnd LAMp COLor Blends the color of all lamps.

BLEnd LAMp BRighTness Blends the brightness of all lamps.

FETch INDicated LAMp COLor

FETch INDicated LAMp BRighTness

SET INDicated LAMp COLor

KEYSTROKES

SET INDICated LAMp BRIGhtness

SET COLor ALL LAMps

SET BRIGhtness of ALL LAMps

For the KNOt commands to work, the knot which is to be altered must be indicated by the cursor.

KNOt INWard

If the knot is a cusp, subsequent speeds or specified slopes will apply to the inward side of the cusp.

KNOt OUTward

If the knot is a cusp, subsequent speeds or specified slopes will apply to the outward side of the slope.

KNOt CALculated

The slope (or slopes, for a cusp) will be calculated.

KNOt SPECified <direction>

The direction is used to set the knot's slope.

KNOt REGular

The knot is turned into a non-cusp.

KNOt CUSp

The knot is turned into a cusp.

KNOt SPEEd <number>

The number specifies the knot's speed. A value of 1.0 corresponds to the middle of the speed slider.

TUMble INTerpolated

TUMble SPECified <direction> <direction>

The first direction indicates the east tumble

KEYSTROKES

axis, the second indicates the north tumble axis.

TAKe NUMber of FRAMES <number>

TAKe PREview SIZE FULL

TAKe PREview SIZE HALF

TAKe MOTion BLUR YES

TAKe MOTion BLUR NO

TAKe RAM YES

TAKe RAM NO

TAKe RAM ECOonomy

TAKe SAVe IMAGES YES

TAKe SAVe IMAGES NO

TAKe LOOP LOOP

TAKe LOOP OSCillate

TAKe LOOP NONE

TAKe ANImation DRAWer <filename>

TAKe FOReground <filename>

TAKe BACKground <filename>

TAKe SAVe GLObal SCENE

TAKe PREview

TAKe LOAd GLObal SCENE

TAKe RENDER ALL

KEYSTROKES

TAKe DELeTe ALL IMAgeS

TAKe CURrent FRAMe <number>

TAKe KEY frame YES

TAKe KEY frame NO

TAKe DURation <number>

TAKe IMAge DRAwer <filename>

TAKe SCENE DRAwer <filename>

TAKe SAVe KEY FRAMe SCENE

TAKe RENder FRAMe

TAKe DELeTe CURrent IMAge

TAKe LOAd KEY FRAMe SCENE

TAKe FRAMe BUFFer <filename>

TAKe NO FRAMe BUFFer

TAKe FRAMe CONtroller <filename>

TAKe NO FRAMe CONtroller

SKY GRAduated <rgb> <rgb>

The first color is the zenith color and the
second is the horizon color.

SKY SOLId <rgb>

SKY NONe

GRound NONe

GRound SOLId <rgb>

KEYSTROKES

GROund CHEckered <rgb> <rgb>

GROund SCALE <number>

The number is the size of the checkerboard squares.

ILLumination <rgb>

The final group of script commands perform functions related to the cursor, the keyboard, and the Tri-View.

ROTate CLOckwise <number>

ROTate COUnterclockwise <number>

WINdow North

Select the northward window. The window commands are important if you plan to use the ROTate commands.

WINdow South

WINdow East

WINdow West

WINdow Up

WINdow Down

TRI-view SIZE <number> <number> <number>

Sets the size of the block of space that the Tri-View represents. The first number is the size in the eastward direction, the second in the northward direction and the third in the upward direction. No size can be less than one tenth of the largest size.

REFRESH

Refresh the Tri-View.

KEYSTROKES

(<number>,<number>,<number>)

Move the cursor to the specified absolute position.

[<number>,<number>,<number>]

Move the cursor to a certain offset from its current position.

(<number>,<number>,<number>)-

(<number>,<number>,<number>)

Create a vertex at each location, unless one already exists. Join them with an edge.

Edges built this way may be chained into any length of curve. If the first and last vertices in a list of joined vertices match, then the list describes a closed loop. A closed loop of only three edges creates a face.

MACRO <character> <script commands> MEND

This command associates the sequence of script commands with the specified character. When the key corresponding to the character is pressed, the commands are executed. If any hot keys are set up with the ALT key, macros will replace them. New hot keys will override macros for the same key. No modifier key, such as the Amiga keys, the ALT keys or the shift keys may be used when defining or using macros. If a font is loaded, it will always take precedence over hot keys and macros. After a font is unloaded, macros and hot keys will start working again.

Because the file "startup.script" is always executed when Sculpt Animate is started, you can use this file to change the default settings to customize your system and install personal macros.

KEYSTROKES

Batch mode

One additional automated method of operating the program has been provided. Execute the command PROJECT BATCH and an empty scene file list will pop up. You can repeatedly click the ADD FILE gadget and select a file to render in the file requester that will appear. No more than 14 files may be selected. Click on a file's name in the batch list to remove it. When you click OK, the program will begin rendering an image for each scene. This command is convenient if you want to make several images and you do not wish to remain near the computer. Alternatively, you may prepare a script file to do the same thing.

Conclusion

Now that you have mastered Sculpt Animate you can use the reference manual in the appendix if you forget how a particular command works.

The scenes that you can build with Sculpt Animate are limited only by your imagination and the amount of memory on your Amiga. Enjoy the product and enrich the world with your creations!

APPENDIX A

SCULPT ANIMATE FUNCTIONAL REFERENCE

The geometry of Sculpt Animate

The Tri-View is a rectangular block of space that is represented by three windows showing wire frame projections of the contents of the Tri-View. The size and shape of the Tri-View may be changed.

The geometry of surfaces is defined by vertices, edges and faces. A vertex is a single point in space. It may be selected or unselected. Vertices are never visible in the images, but are displayed in the Tri-View either as purple for unselected vertices or yellow for selected vertices.

An edge may be created between two vertices. An edge is displayed in the Tri-View as a dark line. Edges are not displayed in images, except in wire frame mode. An edge is considered to be selected if both its vertices are selected.

A face is created whenever three vertices are joined by three edges so each face is triangular. Faces are not explicitly shown in the Tri-View, but they are displayed in the images generated by Sculpt Animate. Faces have a specified color, texture and smoothness. The color is defined by the reflection coefficients for red, green and blue light. The coefficients range from zero to one hundred percent. The texture may be either dull, shiny, mirror, luminous, glass, or metal. If a face has the smoothness property, it is curved so as to blend with adjacent faces.

The location of the observer in the Tri-View is shown by a small blue circle. The position of the observer's target is shown by a small blue x. The position of any lamps are shown by a small white star.

FUNCTIONAL REFERENCE

The Tri-View Gadgets

DRAG BAR GADGET

This is a standard Intuition gadget located in the upper border of each window. It is used to drag the window to a new location on the screen.

THE BACK GADGET AND THE FRONT GADGET

These are standard Intuition gadgets that change the depth arrangement of windows on the screen.

THE REVERSE GADGET

This reverses the viewing direction of a Tri-View window. The first window is normally a downward view. Pressing this gadget will make it an upward view. This is achieved by replacing the contents of the window by its mirror image, thus the first window will have east on the left and west on the right.

THE MOVE TRI-VIEW GADGETS

These gadgets are located in the center of each border and cause the Tri-View to move in the indicated direction. The effect is to make the contents of the Tri-View move in the opposite direction. Holding down the left Amiga key while using a Move Gadget causes the size of the move to be smaller than usual. If the right Amiga key is depressed, the Move Gadget moves the Tri-View further than usual.

THE EXPAND TRI-VIEW GADGET

This gadget causes the Tri-View to expand. If it is used with the left Amiga key, the expansion is small. If it is used with the right Amiga key, the expansion is large. The expansion can be made larger or smaller still by using the left or right ALT keys instead of the Amiga keys.

THE SIZING GADGET

This is a standard Intuition gadget for changing the size of a Tri-View window. Since each window represents a block of space, changing one window's size will result in one or more other windows changing size also.

THE CONTRACT TRI-VIEW GADGET

This gadget causes the Tri-View to contract. If it is used with the left Amiga key, the contraction is small. If it is used with the right Amiga key, the contraction is large. The contraction can be made larger or smaller still by using the left or right ALT keys instead of the Amiga keys.

THE CENTER TRI-VIEW GADGET

This gadget causes the Tri-View to move so that the cursor is left in the center of the Tri-View. If the gadget is used while the left Amiga key is held down, the Tri-View windows are re-sized to their default size. If the gadget is used in conjunction with the right Amiga key, the Tri-View is centered on the default origin.

FUNCTIONAL REFERENCE

THE GRABBER GADGET

This gadget can be toggled on or off. If the Grabber is on, all the selected vertices will move when the cursor is moved. The Grabber is always turned off when the right mouse key is pressed to pull down a menu. When the Grabber is active, the icon for the cursor shows a small Grabber icon on the upper crosshair. If the Grabber is used when no vertices are selected, and the cursor indicates a lamp, then the Grabber can be used to move the lamp.

THE EDGE BUILDING GADGET

If either two or three vertices are selected, then edges are constructed between the selected vertices. If nothing happens when you use this gadget, except that the screen flashes at you, it means that there are more than three selected vertices. Perhaps the extra vertices are outside the Tri-View, or you may have two vertices occupying the same position, both selected.

THE SELECT GADGET

This gadget changes the selection state of all vertices within the Tri-View. If one or more vertices in the Tri-View are selected, then they will be unselected. If no vertices in the Tri-View are selected, then all the vertices in the Tri-View will be selected. In either case, the state of vertices outside the Tri-View are unchanged.

THE ROTATE GADGETS

These gadgets cause selected vertices to rotate about the cursor in the indicated direction. Normally the vertices are rotated by 5 degrees for each click on the gadget. If the left Amiga key is held down, then the rotation is by one degree.

If the right Amiga key is held down then the rotation is by 45 degrees. If the gadget is used with the left ALT key the rotation is 0.1 degrees. If it is used with the right ALT key, the rotation is 180 degrees.

NON-STANDARD USE OF THE MOUSE BY THE TRI-VIEW

When the left mouse button is pressed down, and the pointer is over a Tri-View window, then the current tool is dragged by the mouse. Special actions take place when a specific tool is picked, see the descriptions for TOOLS/SELECTOR, TOOLS/DESELECTOR, TOOLS/MAGNET, TOOLS/CURVE, TOOLS/EXTRUDE, TOOLS/EDGE MAKER and TOOLS/GRABBER for more details. When the default tool is active, the cursor is represented by a blue cross and the following actions happen:

If the left mouse button is pressed, the cursor is moved to the pointer position. If the left button is held down while the mouse is moved, then the cursor will move.

If the right mouse button is pushed while the left button is held down, then a new vertex is created at the cursor position.

If the cursor is moved very close to a vertex in one window of the Tri-View, and the left mouse button is double clicked, the vertex will be selected if it was unselected, or vice versa. The cursor will snap to the location of the vertex in the other two windows.

If an image is being displayed on the screen, it may be pushed behind the Tri-View screen by first clicking on the image. The pointer will disappear. If the right mouse button is pressed, the image will be pushed behind the Tri-View screen.

FUNCTIONAL REFERENCE

Menu commands

The menu commands are shown below. Each command is broken into two or three parts with a slash character (/). The first part is the menu heading, the second part is the menu item, and the third part, if present, is the menu subitem.

PROJECT/LOAD/SCENE

This command invokes a file requester that displays the names of files that contain scenes. See the description of file requesters. If a scene file is selected, Sculpt Animate opens the file and a requester pops up asking which portions of the file are to be loaded. A scene may contain data on objects, lamps, the observer or the world. Click on the boxes to toggle them between YES and NO to specify which parts of the file are to be loaded. Click on the OK button to start loading. A bow is also available to let you specify if you want objects in the scene to be loaded in a selected or unselected state.

Any objects or lamps that are loaded will be added, preexisting lamps or objects will be retained. If an observer or world is loaded, these attributes will override any existing observer or world attributes.

PROJECT/LOAD/IMAGE

This command invokes a file requester and displays the names of files that contain images. If a file is selected, the image is displayed.

PROJECT/LOAD/OBJECT

This command invokes a file requester and displays the names of files that contain scenes. If a file is selected, any objects that are found in the file are loaded at a position relative to the cursor.

PROJECT/LOAD/SCRIPT

This command invokes a file and displays the names of script files. Script files may be prepared with a text editor and contain commands that Sculpt Animate can execute. The command syntax is described in Chapter 14. If a file is selected, the script commands are executed.

PROJECT/LOAD/NAMED OBJECT

This command invokes a hierarchy requester. If a name is selected, then a scene file requester is displayed. If a file is selected then the objects in the scene are loaded and added into the hierarchy at the location chosen in the hierarchy requester.

PROJECT/LOAD/TAKE

A file requester is invoked that lets you select a Take.

PROJECT/LOAD/ANIMATION

A file requester is invoked that lets you select an animation. If you select a file, the animation is loaded into memory and may be displayed with the command .

PROJECT/SHOW/ANIMATION

The current animation is displayed.

PROJECT/LOAD/FONT

A file requester is invoked that lets you select a font. If a font is selected, it is loaded and subsequent keystrokes will cause appropriate objects to be loaded, if the font contains objects corresponding to the keystrokes. The objects are loaded into the current window and the cursor is moved by an amount equal to the size of the object.

FUNCTIONAL REFERENCE

PROJECT/LOAD/CODE

If this command is executed, an attempt will be made to keep all the program code in memory. If free memory falls below a prescribed minimum, then some code will be unloaded.

PROJECT/LOAD/WORKBENCH

An attempt will be made to load the Amiga Workbench.

PROJECT/SAVE/SCENE

This command causes the current scene to be written to the disk. A file requester pops up to solicit the name of a file. If a file name is chosen, a requester will then pop up asking which parts of a scene are to be saved. You may click on the boxes to toggle them from YES to NO. In addition you may specify if all objects are to be saved, or only the selected objects. Click on the box to toggle it between ALL and SELECTED. If the OK gadget is selected, the specified parts of the scene will be written to the file that was named earlier.

PROJECT/SAVE/IMAGE

This command is only available if an image has been rendered or loaded. It causes the current image to be written to the disk. A requester pops up to solicit the name of a file.

PROJECT/SAVE/OBJECT

This command causes selected objects in the current scene to be written to the disk. A requester pops up to solicit the name of a file. When the objects are written to the disk, their positions are taken to be relative to the cursor.

PROJECT/SAVE/NAMED OBJECT

This command invokes a hierarchy requester. If you select a name, then a file requester will pop up. After selecting a file, then objects with the chosen name are saved.

to the scene file with hierarchy names derived by stripping off the parent of the current name.

PROJECT/SHOW/IMAGE

If an image is loaded, then it will be displayed on the screen. The image can be pushed back by pointing at the image and first clicking with the left mouse button and then clicking with the right mouse button.

PROJECT/SHOW/ANIMATION

If an animation is loaded, it is displayed. The animation can be stopped by clicking with the right mouse button or by pressing the escape key. The speed of the animation can be changed by pressing the keys 0 to 9.

PROJECT/SHOW/PREVIEW

If an animation is loaded into memory, it is displayed. The preview can be stopped by clicking with the right mouse button or by pressing the escape key. The speed of the animation can be changed by pressing the keys 0 to 9. When a preview is stopped, you will be asked if you wish to unload it.

PROJECT/UNLOAD/IMAGE

If an image is loaded, it is removed, and the memory that it used is made available for other purposes.

PROJECT/UNLOAD/ANIMATION

If an animation is loaded, it is removed, and the memory that it used is made available for other purposes.

PROJECT/UNLOAD/PREVIEW

If a preview is loaded, it is removed, and the memory that it used is made available for other purposes.

FUNCTIONAL REFERENCE

PROJECT/UNLOAD/FONT

If a font is loaded, it is unloaded and keys resume their normal operation as hot keys or macros.

PROJECT/UNLOAD/CODE

All unneeded code is removed from memory until it is needed. As soon as code is no longer needed, it is again unloaded.

PROJECT/UNLOAD/WORKBENCH

An attempt is made to remove the Amiga Workbench. This will fail if a CLI window is present.

PROJECT/BATCH

A requester pops up that allows you to select one or more scene files. Images are then rendered for each of the selected files.

PROJECT/ABOUT

The version number and other information about Sculpt Animate is displayed.

PROJECT/QUIT

After verifying that this is your intention, Sculpt Animate stops operating and returns control to the workbench or CLI.

EDIT/SELECT/ALL

All vertices, and hence all edges and faces, become selected.

EDIT/SELECT/CONNECTED

All vertices that are connected, directly or indirectly, to the indicated vertex become selected. If a vertex lies within a few pixels of the cursor in the active window of the Tri-

FUNCTIONAL REFERENCE

View, this vertex is considered the indicated vertex. If more than one vertex meets this criteria and they are distributed in depth, then the indicated vertex is the one that is closest to the cursor in the depth direction.

EDIT/SELECT/INDICATED VERTEX

The vertex that is indicated by the cursor is selected.

EDIT/SELECT/SWAP

All selected vertices become deselected and vice versa.

EDIT/SELECT/NAMED VERTICES

A hierarchy requester will be invoked. After selecting a name, if the OK button is pressed, all the named vertices will be selected.

EDIT/SELECT/INDICATED EDGE

If an edge is indicated by the cursor, then the two vertices that define the edge are selected.

EDIT/SELECT/INDICATED SPLINE

If the cursor indicates a vertex that is a part of a spline, then all the knots on the spline are selected.

EDIT/SELECT/INDICATED PATH

If the cursor indicates a path, then all the vertices on the path are selected.

EDIT/DESELECT/ALL

All vertices, and hence all edges and faces become deselected.

FUNCTIONAL REFERENCE

EDIT/DESELECT/CONNECTED

All vertices that are connected, directly or indirectly, to the indicated vertex become deselected. If a vertex lies within a few pixels of the cursor in the active window of the Tri-View, this vertex is considered the indicated vertex. If more than one vertex meets this criteria and they are distributed in depth, then the indicated vertex is the one that is closest to the cursor in the depth direction.

EDIT/DESELECT/INDICATED VERTEX

The vertex that is indicated by the cursor is selected.

EDIT/DESELECT/SWAP

All selected vertices become deselected and vice versa.

EDIT/DESELECT/NAMED VERTICES

A hierarchy requester will be invoked. After selecting a name, if the OK button is pressed, all the named vertices will be deselected.

EDIT/DESELECT/INDICATED EDGE

If an edge is indicated by the cursor, then the two vertices that define the edge are deselected.

EDIT/ERASE/SELECTED VERTICES

All selected vertices are erased. Any edges and faces that are defined by these vertices are also erased.

EDIT/ERASE/SELECTED EDGES

All selected edges are erased. Any faces that are defined by these edges are also erased. A selected edge is one that has a selected vertex at each end.

EDIT/ERASE/INDICATED VERTEX

The vertex that is indicated by the cursor is erased.

EDIT/ERASE/INDICATED EDGE

The edge that is indicated by the cursor is erased.

EDIT/ERASE/INDICATED LAMP

The indicated lamp is erased. If a lamp lies within a few pixels of the cursor in the active window of the Tri-View, this lamp is considered the indicated lamp. If more than one lamp meets this criteria and they are distributed in depth, then the indicated lamp is the one that is closest to the cursor in the depth direction.

EDIT/ERASE/ALL LAMPS

All lamps are erased.

EDIT/ERASE/NAMED VERTICES

A hierarchy requester will pop up. If a name is selected and the OK button pressed, all named vertices will be erased.

EDIT/ERASE/INDICATED SPLINE

If the indicated vertex is a member of a spline, then the spline is erased. Only the underlying spline structure is erased, the vertices and edges belonging to the spline remain.

EDIT/ERASE/INDICATED PATH

If the indicated vertex is a member of a path, then the path is erased. Only the underlying path structure is erased, the vertices and edges belonging to the path remain.

FUNCTIONAL REFERENCE

EDIT/ERASE/ALL

All vertices, edges and lamps are erased. If a take is loaded, you will be asked if you want to have the take erased. Erasing a take only removes the take in memory. The copy of the take on disk still remains.

EDIT/MODIFY/FACES

The modify faces requester is invoked. It can be used to change the default color, texture and smoothing for yet to be created faces. It can also examine and change the color, texture and smoothness of selected faces.

EDIT/MODIFY/MODIFY LAMPS

The modify lamps requester is invoked. It can be used to change the default color and brightness that is used for newly created lamps. It can also examine and change the color and brightness of either an indicated lamp or all existing lamps.

EDIT/MODIFY/WIRE-FRAME COLORS

This command invokes a color requester that can be used to modify the background color and the color used to draw edges for wireframe and preview renderings. Click on the button at the top right to toggle between FIELD and LINES, and use the color sliders in the usual manner.

EDIT/MODIFY/LOCAL ORIGIN

A hierarchy requester is invoked. Use the SET button to set the local origin to the cursor position. Use the LOCK button to lock the local origin onto an indicated vertex. The SHOW button will move the cursor to the local origin.

EDIT/MODIFY/INDICATED KNOT

The modify knot requester is invoked, it can be used to modify the parameters of a spline. See Modify Knot, below.

EDIT/MODIFY/INDICATED TUMBLE

The cursor must indicate a vertex that lies on a path. The Modify Tumble requester is invoked, which can be used to modify the orientation of the tumble axes, see Modify Tumble, below.

EDIT/MODIFY/TAKE

If a Take is loaded, this command invokes the Modify Take requester, see below. This requester acts as a control panel for creating, modifying and rendering animations. EDIT/DO/EXPAND A requester is invoked that can be used to expand the set of selected vertices.

EDIT/DO/SUBDIVIDE

This operation divides each selected edge into two. Additional edges are created to ensure that subdivided faces remain triangulated. The effect is that faces that are selected are split into four, while faces that have two selected vertices are divided into two.

If only two vertices are selected, and they are not joined by an edge, then a requester pops up asking how many vertices to insert. Vertices are uniformly spaced between the two selected vertices and all are linked by edges.

EDIT/DO/SPIN

The spin operation takes place in the active Tri-View window. Each selected edge is swept around the spin axis creating new faces. Each selected vertex is swept around the spin axis creating new edges. The spin axis is horizontal on the screen and passes through the cursor position.

Normally the spin operation is applied to a sequence of connected edges created with the curve tool. It then produces a surface of revolution about the horizontal axis.

You are first asked how many steps are required for the spin operation. Then you are asked for a sweep direction.

FUNCTIONAL REFERENCE

EDIT/DO/FILL

In order for this command to operate, the selected vertices must all lie on a single plane. In addition the selected edges must form one or more closed loops. The loops must not intersect one another. The FILL command constructs new edges so that the interior of each loop is fully triangulated. If one loop lies entirely inside another, then that loop is regarded as a hole for the purpose of filling. For example, if the fill command is applied to two concentric circular loops, a filled annulus will be produced.

EDIT/DO/UNSLICE

This command treats the selected vertices as slices through an object. All the selected vertices must be members of closed loops. Each loop must lie in a plane. Each plane must be parallel and distinct. The loops are connected to approximate the object defined by the slices.

EDIT/DO/REFLECT

The selected vertices and edges are duplicated by reflecting them in a plane defined to be vertical and to pass through the cursor location in the current Tri-View window. Any vertices in the plane of reflection are not duplicated.

EDIT/DO/MAKE HELIX

A helical or spiral object is created by rotating the selected vertices and edges around a spin axis while changing the radius and distance along the axis. The spin axis is horizontal in the current Tri-View window. The radial and axial displacements are determined in one of two ways.

If there is no indicated vertex, the selected vertices must lie on a non-looping curve. The end points of the curve represent radial and axial offsets for one rotational sweep. The object created is a single surface, one turn is connected to the next.

FUNCTIONAL REFERENCE

If there is an indicated vertex, then it must be one end of a non-looping curve called the control curve. Each vertex, except for the first, represents a radial and axial offset for one sweep rotation. Offsets for angles other than multiples of 360 degrees are calculated by smooth interpolation or extrapolation.

In either case you will be prompted for a number of steps per turn and a total number of sweep steps.

EDIT/DO/MAKE SPLINE

All the selected vertices must lie on a single curve. Each vertex on the curve must be joined to two other vertices with edges, except for the two end vertices that are connected to one vertex with an edge. Alternatively, the curve may be a closed loop with every vertex being joined to two other vertices with edges. A spline will be created and the selected vertices will become knots. Small arrowheads will be drawn on the edges near the knots.

EDIT/DO/MAKE PATH

The cursor must be placed on the vertex that is to be the starting point of a path. Vertices connected to this vertex must lie on a single curve or loop. If the path is to be a loop, the second vertex on the path must be the only selected vertex. If these conditions are satisfied, the curve becomes a path.

EDIT/DO/SHOW PATH POSITION

If a Take is loaded, the cursor will be moved to the vertex on an indicated path that corresponds to the path position for the current frame.

EDIT/DO/HIDE SELECTED VERTICES

Any selected vertices become hidden. They do not show in the Tri-View, and they cannot become selected. By hiding vertices, they can be excluded from many operations

FUNCTIONAL REFERENCE

that require selection. Tri-View refresh is faster if most vertices are hidden.

EDIT/DO/REVEAL HIDDEN VERTICES

All hidden vertices are made non-hidden.

EDIT/DO/MAKE TRI-VIEW SMALL

This command resizes the Tri-View windows to their default size.

EDIT/DO/MAKE TRI-VIEW BIG

This command makes the Tri-View windows big enough that they almost fill the screen.

EDIT/ADD/DUPLICATE

Each selected vertex and edge is duplicated. The duplicates are created in an unselected state. The duplicates occupy the same positions as the original, so they are not visible unless the Grabber is used to move the originals to a new location. This command may be used repeatedly as the original object is dragged from one location to another, in order to create multiple copies of the object.

EDIT/ADD/SPHERE

Add a pseudo-spherical polyhedron. A requester asks how many subdivisions of an icosahedron are required. A value of zero yields a regular icosahedron with 20 faces. Each extra subdivision quadruples the number of faces, so storage can easily be exceeded. If smoothing is enabled, even a coarse figure looks quite spherical.

— The size of the sphere is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/HEMISPHERE

Add a polyhedron shaped like a solid hemisphere. A requester asks how many subdivisions are required. The larger the value, the more closely the object resembles a hemisphere.

The size of the hemisphere is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/CUBE

A cube is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/PRISM

A prism is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/DISK

A disk is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/CIRCLE

A circle is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/CYLINDER

A cylinder is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/TUBE

A tube is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

FUNCTIONAL REFERENCE

EDIT/ADD/CONE

A cone is constructed whose size is chosen to fit inside the current Tri-View volume with a small clearance.

EDIT/ADD/LAMP

A lamp is created at the cursor location with the current brightness and color.

EDIT/ADD/VERTEX

A vertex is created at the cursor position. The vertex is selected.

EDIT/ADD/EDGES

This command has the same effect as clicking on the edge builder gadget.

EDIT/SNAP/CURSOR TO VERTEX

The cursor is moved to the location of the nearest vertex.

EDIT/SNAP/CURSOR TO CENTER

The cursor is snapped to the center of the selected vertices.

EDIT/SNAP/CURSOR TO CENTROID

The cursor is snapped to the centroid of the selected vertices.

EDIT/SNAP/CURSOR TO GRID

The cursor is moved to the nearest grid location.

EDIT/SNAP/SELECTED VERTICES TO GRID

Each selected vertex is moved to the nearest grid location.

EDIT/SNAP/CONNECTED VERTICES TO SPHERE

The cursor must be placed over a vertex. The vertices that are connected to this vertex are examined and the size of a rectangular box that would just contain the vertices is determined. The vertices are moved radially from the center of the box until they lie on the surface of a sphere whose diameter is equal to the longest side of the box.

The effect of this command is to make an object approximate a sphere. The approximation will be quite good if the object has many faces.

EDIT/SNAP/SELECTED VERTICES TO PLANE

A plane is calculated that best fits the set of selected vertices. The selected vertices are moved so as to lie in the plane.

EDIT/NAME/SELECTED VERTICES

A hierarchy requester is invoked. After choosing or creating a hierarchy name, the selected vertices are given the current name when the OK gadget is pressed.

EDIT/NAME/CONNECTED VERTICES

A hierarchy requester is invoked. After choosing or creating a hierarchy name, the vertices that are connected to the indicated vertex are given the current name when the OK gadget is pressed.

EDIT/NAME/INDICATED PATH

A hierarchy requester is invoked. After choosing or creating a hierarchy name, the indicated path is given the current name when the OK gadget is pressed.

FUNCTIONAL REFERENCE

EDIT/NAME/INDICATED LAMP

A hierarchy requester is invoked. After choosing or creating a hierarchy name, the indicated lamp is given the current name when the OK gadget is pressed.

EDIT/GRID/ON

A requester pops up asking for a grid size. The grid size should be given in units of the least significant digit of a coordinate display. Unless the grid spacing is less than a few pixels, the grid is drawn in the Tri-View windows.

EDIT/GRID/OFF

The grid display is turned off.

EDIT/COORDINATES

This command causes a coordinate window to pop up. Before the window is displayed, a prompt is given for the location of the decimal point for numerical values. This window can be dragged to any part of the screen and depth arranged. The window can be removed with its close gadget. The window displays the current position of the cursor measured eastward, northward and upward of an origin which can be reset with the X-shaped gadget at the lower right of the window.

In addition, the window also displays the distance between two selected points. These points can be set by using the End Point Gadgets. If an End Point Gadget is pressed, the end point is either the vertex indicated by the cursor, or the current cursor position. If a vertex is specified as an end point, then the end point moves if the vertex is moved. The end points are represented by end point icons in the Tri-View windows. When an end point is locked to a vertex, it is shown in purple, otherwise in blue.

The gadget on the lower left of the window can be used to set an angle marker, or lock it to a vertex, in a similar manner to the end points. The box to the left of the angle

marker shows the angle from one end point marker to the angle marker to the other end point marker.

If you click on one of the three coordinate value boxes, you can enter or edit the value and the cursor will move to the specified location.

TOOLS/SELECTOR

This command picks the selector tool. It is represented in the Tri-View by a yellow square. It represents a cube in space. Any vertices that enter the cube become selected. The tool may be dropped by pressing the right mouse button.

TOOLS/DESELECTOR

This command picks the deselector tool. It is represented in the Tri-View by a purple square. It represents a cube in space. Any vertices that enter the cube become deselected. The tool may be dropped by pressing the right mouse button.

TOOLS/MAGNET/ATTRACT

The magnet tool is selected and is placed in attract mode. The tool is represented by an image of a horseshoe magnet. If the left mouse button is held down and the right mouse button is clicked, all selected vertices are moved towards the magnet. The strength of the attraction falls off inversely as the square of the distance between the vertex and the magnet, except when the vertex is very close to the magnet, when the attraction varies directly with the distance. The magnetic attraction is increased if the magnet is operated with the right Amiga key pressed down. The attraction is reduced if the left Amiga key is down when the magnet is operated. The tool may be dropped by pressing the right mouse button.

TOOLS/MAGNET/REPEL

The magnet tool is selected and is placed in repel mode. The tool is represented by an image of a horseshoe magnet.

FUNCTIONAL REFERENCE

If the left mouse button is held down and the right mouse button is clicked, all selected vertices are moved away from the magnet. The strength of the repulsion falls off inversely as the square of the distance between the vertex and the magnet, except when the vertex is very close to the magnet, when the repulsion varies directly with the distance. The magnetic repulsion is increased if the magnet is operated with the right Amiga key pressed down. The repulsion is reduced if the left Amiga key is down when the magnet is operated. The tool may be dropped by pressing the right mouse button.

TOOLS/CURVE

The curve tool resembles the cursor crosshairs except that the upper and right portions of the cursor are replaced by an icon of a pencil. When the curve tool is picked and the left mouse button is held down and the right mouse button is clicked, a new vertex is created and an edge is constructed from this vertex to the previous vertex, except if this is the first vertex made with this invocation of the curve tool. Prior to creating an edge, a rubber band line is drawn from the previous vertex to the cursor so as to indicate where an edge could be created. If the curve tool is indicating an existing vertex when the right button is clicked, then the previous vertex will be joined to the indicated vertex with an edge. The tool may be dropped by pressing the right mouse button or by the act of building an edge to an indicated vertex.

TOOLS/EXTRUDE

The extrude tool duplicates the selected object and joins the duplicate and the selected object. Edges are linked to create new faces only if the edge is a part of not more than one face. This means that 'interior' edges do not extrude into faces. The Grabber tool is activated. If you move to another window and move the cursor, the selected object is dragged, while the duplicate is fixed. The effect is to extrude a two dimensional shape into a three dimensional solid. Remember to drop the Grabber when the extrusion is complete.

TOOLS/EDGE MAKER

The edge maker is used to join existing vertices with edges. A vertex is marked by moving the cursor to the vertex and while holding down the left mouse button, clicking with the right mouse button. The first and second marked vertices are joined by an edge, the third and fourth, etc. Prior to creating an edge, a rubber band line joins the previous vertex to the cursor to indicate the location that an edge would have.

TOOLS/GRABBER

This command has the same effect as clicking on the Grabber gadget in a Tri-View window.

OBSERVER/MODE/WIRE FRAME

This command sets the observer in wire frame mode. Rendered images only show edges and not faces. The perspective is correct.

OBSERVER/MODE/SKETCH

This command sets the observer in SKETCH mode. Sketch mode is similar to painting mode, described below, except that rendering is faster, but intersecting faces are not treated correctly and sometimes a face is drawn in front of another face when it should be drawn behind.

OBSERVER/MODE/PAINTING

This command sets the observer in painting mode. A painting is an image where each face is represented by a single color and no shading is portrayed across a face. Shadows are not rendered. The image is produced rapidly, and no significant speed improvement is achieved if the image size is reduced.

FUNCTIONAL REFERENCE

OBSERVER/MODE/SCANLINE PAINTING

This sets the observer to use a variation of painting mode. The image will be rendered more quickly for complicated scenes.

OBSERVER/MODE/SCANLINE SNAPSHOT

This sets the observer to use another variation of snapshot mode, described below, except that reflections will not be shown. The image will again be rendered more quickly for complicated scenes. Full 4096 color images are produced.

OBSERVER/MODE/FLASHSHOT

This command sets the observer in snapshot mode. A snapshot is a ray traced image that excludes the effects of shadows. The time taken to produce an image is proportional to the image size, so small images can be produced much more quickly than large images.

OBSERVER/MODE/PHOTO

This command sets the observer in photo mode. A photo is a ray traced image that includes the effects of shadows. The time taken to produce an image is proportional to the image size, so small images can be produced much more quickly than large images. Photos take longer to generate than snapshots.

OBSERVER/MODE/LO-RES

This specifies that images are to be made in Amiga low resolution mode with approximately 320 pixels across the screen.

OBSERVER/MODE/HI-RES

This specifies that images are to be made in Amiga high resolution mode with approximately 640 pixels across the

screen. If Hold and Modify (HAM) graphics are employed, low resolution is used.

OBSERVER/MODE/NO-INTERLACE

This command specifies that images are to be produced without interlace, so that approximately 200 pixels are used in the vertical direction.

OBSERVER/MODE/INTERLACE

This command specifies that images are to be produced with interlace, so that approximately 400 pixels are used in the vertical direction. Some screen flicker may be perceived with a short persistence monitor.

OBSERVER/MODE/BITPLANES

This command enables you to set up a smaller than normal number of colors for non-HAM rendering. The number of colors is two raised to the power of the number of bitplanes that you specify. If you request 24 bitplanes, then no image is produced, but image data is written to three files. You will be asked to specify screen and pixel size.

OBSERVER/LOCATION

The location of the observer is moved to the cursor position. The observer's location is portrayed in the Tri-View by a small blue circle.

OBSERVER/TARGET

The observer's target is moved to the cursor position. The observer's target is portrayed in the Tri-View by a small blue x. The target position will be in the center of any images that are produced.

OBSERVER/LENS/NORMAL

This command sets the width of view to correspond to that of a 50mm lens on a 35mm camera.

FUNCTIONAL REFERENCE

OBSERVER/LENS/WIDEANGLE

This command sets the width of view to correspond to that of a 28mm lens on a 35mm camera.

OBSERVER/LENS/TELEPHOTO

This command sets the width of view to correspond to that of a 135mm lens on a 35mm camera.

OBSERVER/LENS/SPECIAL

This will invoke a requester for the focal length of a non-standard lens. Enter the value in millimeters. A small value, like 20, corresponds to a wide view, whereas a large number, like 100, corresponds to a narrow view where distant objects seem closer.

OBSERVER/EXPOSURE/AUTO

This command instructs Sculpt Animate to determine the screen brightness automatically.

OBSERVER/EXPOSURE/MANUAL

This command overrides the automatic exposure setting. Numbers larger than 100 cause the image to be brighter, values less than 100 cause the image to be darker.

OBSERVER/IMAGE SIZE/TINY

This command causes subsequent images to be made that are 1/8 of the full screen size.

OBSERVER/IMAGE SIZE/SMALL

— This command causes subsequent images to be made that are 1/4 of the full screen size.

OBSERVER/IMAGE SIZE/MEDIUM

This command causes subsequent images to be made that are half of the full screen size.

OBSERVER/IMAGE SIZE/FULL

This command causes subsequent images to be made that are the full screen size.

OBSERVER/IMAGE SIZE/JUMBO

This command causes subsequent images to be made that are larger the full screen size.

OBSERVER/IMAGE SIZE/VIDEO

This command causes subsequent images to be made that are taller the jumbo screen size.

OBSERVER/TILT

A requester is summoned to obtain an angle that the image is to be rotated. A value of 90 can be used to produce vertical format images.

OBSERVER/ANTI-ALIASING/NONE

No anti-aliasing is used for maximum rendering speed.

OBSERVER/ANTI-ALIASING/GOOD

A simple anti-aliasing scheme is employed that does not incur a large penalty in computing time.

OBSERVER/ANTI-ALIASING/BEST

An anti-aliasing scheme is employed that will incur a moderate penalty in computing time.

FUNCTIONAL REFERENCE

OBSERVER/ANTI-ALIASING/DITHERING

A requester is invoked that asks you what percentage of standard dithering is required. Values in the range 0 to 200 are useful.

OBSERVER/DISPLAY/EARLY

Images are usually not displayed until they are fully rendered. This setting declares that images should be displayed as rendered, if system resources permit.

OBSERVER/DISPLAY/LATE

This re-instates the normal delayed image display.

OBSERVER/START

This command initiates image generation. When a photo or a snapshot is being generated, the grinding gear wheel icon vanishes after a short time and all Sculpt Animate commands (except OBSERVER/START) may be executed without changing the image as it is produced. The image screen can be pushed behind the Tri-View screen by clicking on the image first with the left button and then with the right button.

OBSERVER/ABORT

This command will terminate any image generation.

OBSERVER/STATUS

This command will enable or disable the status line at the top of the screen. It shows the following information.

- The amount of unused chip memory
- The amount of unused fast memory
- The number of selected vertices

The number of unselected vertices
The number of edges
The number of faces
The number of lamps

WORLD/SKY

This command invokes a sky requester, which closely resembles the color requester. The gadget at the top is a three way toggle that can take the values NONE, SOLID or GRADUATED. Click on the gadget to change its value. When NONE is selected, no sky is generated, and the color sliders are not used. If SOLID is selected, then a single color sky is produced, and the color sliders can be used to specify the color. The SOLID color is stored for use in reflections and glass if the sky is subsequently set to NONE. If GRADUATED is selected, the swatch of color at the lower center of the requester splits in two. The upper half is the sky color at the zenith (directly overhead), while the lower half represents the color at the horizon. The two colors will be blended at intermediate elevations from the horizon. Either the upper or lower half of the swatch is emphasized with a three sided border. Click on the swatch to toggle the selection between the two halves. The color of the selected half can be changed with the sliders.

Note that when a non-HAM rendering mode is specified, the sky is displayed in a single color.

WORLD/GROUND

This command invokes a ground requester, which closely resembles the color requester. The gadget at the top is a three way toggle that can take the values NONE, SOLID or CHECKERED. Click on the gadget to change its value. When NONE is selected, no ground is generated, and the color sliders are not used. If SOLID is selected, then a ground plane is created and the color sliders can be used to specify the ground color. If CHECKERED is selected, the swatch of color at the lower center of the requester splits in

FUNCTIONAL REFERENCE

two. This represents the two colors of a checkerboard patterned ground plane. Either the upper or lower half of the swatch is emphasized with a three sided border. Click on the swatch to toggle the selection between the two halves. The color of the selected half can be changed with the sliders.

When CHECKERED is selected, the gadget resembling a checkerboard at the top right of the requester becomes active. It is called the Checkerboard Scale Gadget. If this gadget is clicked, the size of squares that are generated on the ground is set equal to the longest dimension of the Tri-View volume.

If either a SOLID or CHECKERED ground is chosen, then a line representing the ground is drawn in two of the Tri-View windows.

WORLD/ILLUMINATION

This command invokes a color requester that can be used to set the brightness and color for background illumination. This light source bathes the entire scene without casting shadows. Normally the V value should be set at about 1/4 to prevent shadows from being totally black. Too high a value, or the absence of lamps will result in a flat and lifeless image.

The file requester

This requester is used to select new or existing files. It has three main panels. The panel on the upper right lists the currently mounted disk volumes. To select a new volume, double click on the volume that you need. If more than six volumes are present, the scroll bar or the scroll gadgets can be used. The currently selected volume is highlighted.

The panel on the upper left works in a similar manner, but shows the drawers on the current volume. Either double

click on drawer, or type in the name in the box immediately above the panel. The first item in the panel is marked '/ Parent Directory' and is used to select the drawer containing the current list of drawers. Using this panel you can traverse a set of nested drawers.

The panel on the lower left is used to select files once the volume and drawer have been selected. Again double click on the file or enter a name in the box above the panel. To create a new file, of course, you must enter its name.

Additional buttons allow you to rename or delete files. The OK button can be used to select a file, if the desired file has been highlighted by clicking on it.

The hierarchy requester

Sculpt Animate makes use of a hierarchy of names to represent complicated objects and composite motions. The names are linked to one another in a tree-like structure. In the requesters that use the hierarchy, a name can be linked to one name to the left. The name to the left is called a parent, by analogy to a family tree. A name can also be linked to many names on the right, these names are called offspring.

Because the hierarchy structure could be very large, only a portion of it is displayed at a given time. Up to nine names that share a common parent are displayed one above another. The name in the center of the list is the 'current name', i.e. the name under consideration. You can click on a name to make the list move so that that name becomes the current name. Also you can use the scrolling gadgets to select the current name.

A new name can be added to the current generation by clicking on the box titled 'Add name:' and typing in a new name. When you press return, the name will be inserted into the current generation in alphabetic order. It does not matter if names are entered in lower case or upper case, 'JOE' is regarded as the same as 'joe'.

FUNCTIONAL REFERENCE

The gadget marked OFFSPRING can be used to move to the right in the tree. Any names added will then be the offspring of the previous current name. The box titled 'Parentage' displays a list of parents, grandparents, etc. of the current name. If the list of ancestors is too big to fit in the box, three periods will be shown to indicate that the list is incomplete. If you click on the parentage box, you will move to the left in the hierarchy structure, and the parent of the current name will become the new current name.

Once a name has been established, it can be given to one or more vertices. A vertex cannot have more than one name at a time. Giving it a new name will mean that it no longer responds to the old name. Names can also be given to paths, in which case all the vertices in the path will have the same name. Names can also be given to lamps.

Names can be erased from the hierarchy with the ERASE NAME gadget, in which case any objects having that name will no longer be named.

When a global scene is loaded as a part of a frame, any path in the hierarchy is treated as a request to move and rotate the offspring that are attached to the path.

The hierarchy can be edited in a normal manner with the gadgets CUT PASTE and COPY.

The expansion requester

This requester can be used to expand the set of selected vertices. The expansion (or contraction) is uniform in the selected directions and is directed away from either the selected vertices CENTER or CENTROID, or the CURSOR, as set by a gadget in the requester's title.

The expansion can take place in the north-south, east-west, or up-down direction, or any combination selected by buttons on the requester. Expansion may be done in discrete steps of 2 times or 1/2, by pushing the appropriate buttons. Alternatively the upper slider may be used. Moving it to the

right of center causes expansion, left of center causes contraction. The lower slider may be used for fine adjustment. It is 50 times less sensitive.

The expansion requester may be dragged to a new location so that it does not obstruct the Tri-View. The drag bar is at the top of the requester. If the CANCEL button is pressed, the vertices revert to their positions before the expansion was attempted.

Color requesters

Several requesters allow you to select a color. The color may be changed by using sliders to set the red, green and blue values. Alternatively, sliders can control the hue, saturation and value. Hue varies continuously from red to green to blue and back to red. Saturation represents the purity of the color. As the saturation is decreased, more white is mixed in. The value is a measure of the brightness of the shade.

The modify faces requester

This requester is used to modify the current face color, texture and smoothness. The color sliders work as described above. The smoothing value can be toggled between ON and OFF by clicking on the gadget. The texture can be cycled through the range of possible values: DULL, SHINY, MIRROR, LUMINOUS, GLASS and METAL by clicking on the gadget. The cycle direction can be reversed by holding down the ALT key while clicking on the gadget.

The face properties of selected faces can be modified by pressing the appropriate SET gadgets. The face properties of selected faces can also be extracted, if the properties of all selected faces are the same, with the FETCH gadgets. In the case of color, an average color can be extracted even when the colors of selected faces differ, by the BLEND gadget.

FUNCTIONAL REFERENCE

The modify lamps requester

This requester is used to modify the current color and brightness of lamps. If a lamp is indicated by the cursor, then the gadget marked WHICH LAMP can be toggled between ALL, meaning that all lamps are to be considered, and INDICATES which means that only the indicated lamp is to be considered.

Colors can be set in the usual way with the sliders. FETCH and BLEND gadgets are provided for extracting the color and brightness, while SET gadgets can be used to change the properties of lamps.

The modify knot requester

This requester can be used to modify the parameters of a knot. It has the following gadgets:

Knot type:

This gadget can be toggled between REGULAR and CUSP. If the knot is a cusp, the knot has separate parameters with regard to the curve on either side of the knot. The side with the arrowhead pointing towards the knot is called the inward side and the other side, with the arrowhead pointing away from the knot, is called the outward side.

Side:

If the knot is a cusp, this gadget can be toggled between INWARD and OUTWARD. The side refers to which side of the curve is to be modified by the parameters.

Slope:

This gadget can be toggled between CALCULATED and SPECIFIED. If the gadget is set to CALCULATED, then the slope of the spline is determined from the positions of neighboring knots. If the slope is set to SPECIFIED, then

the slope can be adjusted by the Tri-View rotate gadgets, while the requester is displayed. The slope is indicated by a yellow arrow.

Speed

This parameter is controlled by a slider. Small values will cause non-knot vertices to cluster near the knot, and for the curve to deviate quickly from the tangent.

CANCEL

This gadget removes the requester and returns the knot parameters to their values before the requester was invoked.

OK

This gadget causes the requester to be removed and the knot parameters to be accepted.

The modify tumble requester

When this requester is invoked, the tumble axes will be displayed as follows: The east tumble axis will be shown by a yellow line, the north tumble axis by a white line and the up tumble axis by a purple line. The tumble axes can be rotated by the Tri-View rotate gadgets.

The requester provides the following gadgets:

MATCH LAST

If you click on this gadget, the tumble axes are aligned to be parallel with the tumble axes that were examined most recently.

Method:

This gadget can be toggled between INTERPOLATED and SPECIFIED. This gadget must be set to SPECIFIED to be able to rotate the tumble axes. If it is set to

the slope can be adjusted by the Tri-View rotate gadgets, while the requester is displayed. The slope is indicated by a yellow arrow.

Speed

This parameter is controlled by a slider. Small values will cause non-knot vertices to cluster near the knot, and for the curve to deviate quickly from the tangent.

CANCEL

This gadget removes the requester and returns the knot parameters to their values before the requester was invoked.

OK

This gadget causes the requester to be removed and the knot parameters to be accepted.

The modify tumble requester

When this requester is invoked, the tumble axes will be displayed as follows: The east tumble axis will be shown by a yellow line, the north tumble axis by a white line and the up tumble axis by a purple line. The tumble axes can be rotated by the Tri-View rotate gadgets.

The requester provides the following gadgets:

MATCH LAST

If you click on this gadget, the tumble axes are aligned to be parallel with the tumble axes that were examined most recently.

Method:

This gadget can be toggled between INTERPOLATED and SPECIFIED. This gadget must be set to SPECIFIED to be able to rotate the tumble axes. If it is set to

FUNCTIONAL REFERENCE

INTERPOLATED, the tumble axes will be calculated from the orientation of adjacent specified tumble axes.

NEXT

This gadget will cause the cursor to be moved to the next vertex on the path. The cursor will also be centered in the Tri-View.

PREVIOUS

As above, except that the previous vertex on the path is chosen.

The modify Take requester

This requester can be used to change the parameters of the Take and to initiate various actions. The gadgets that are available, in some cases depends upon the settings of other gadgets. The following gadgets are available:

Take mode:

This gadget is always available, it can be toggled between FRAME and GLOBAL. When it is set to GLOBAL, only gadgets that relate to the animation as a whole will be accessible. When it is set to FRAME, then only gadgets that are relevant to a particular frame will be available.

Number of frames:

This gadget can be used to enter the total number of frames that are needed by the animation. RAM animation: This gadget can be cycled through YES, NO and ECONOMY. The ECONOMY style of RAM animation is faster to produce, but employs less compression.

Loop mode:

This gadget can be cycled through NONE, LOOP and OSCILLATE. This gadget is only available when the RAM animation gadget is not set to NO. If NONE is specified, then any resulting animation will play once and then stop. If LOOP is requested, then the animation will be repeated until it is stopped manually. If OSCILLATE is set, the animation will first be played forwards and then backward. This sequence is repeated until the animation is stopped.

Frame controller:

This gadget is only available when RAM animation is set to NO. The gadget can be set to YES or NO. When it is set to YES, a file requester will be invoked to let you specify a driver for a single frame controller which serves as an interface between the computer and an output device such as a video tape recorder or a film recorder.

Frame buffer:

This gadget is only available when RAM animation is set to NO. The gadget can be set to YES or NO. When it is set to YES, a file requester will be invoked to let you specify a driver for a frame buffer which is a hardware device that displays an image or produces a video signal.

Preview size:

This gadget can be toggled between FULL and HALF and it controls the size of the wire frame preview display, when the PREVIEW button is pressed.

Save images:

—This gadget can be toggled between YES and NO. Normally it is set to YES and all images that are rendered are saved to disk. If the gadget is set to NO, then the images may still be saved to disk, but they will be deleted as soon as possible.

FUNCTIONAL REFERENCE

Motion blur:

This gadget can be toggled between YES and NO. If motion blur is specified, then moving objects will seem to be blurred slightly along the direction of motion.

Foreground:

This gadget can be used to enter the name of an image that is to be treated as a foreground. The image should have areas that are transparent, so that the animated scene is visible. Transparency is specified by setting a pixel to color zero, or by setting a pixel in HAM mode to black. Such images can be prepared with a paint program. The actual file name must end with '.image', but this extension should not be entered into the gadget. The foreground image should be the same size as the image being rendered.

Background:

This gadget can be used to enter the name of an image that is to be treated as a background. When an image is rendered and the sky is turned off, any part of the image that would otherwise be recorded as sky will be replaced by the background image. The actual file name must end with '.image', but this extension should not be entered into the gadget. The foreground image should be the same size as the image being rendered.

SAVE GLOBAL SCENE

This gadget causes the objects in the Tri-View to be saved as the global scene for the Take.

LOAD GLOBAL SCENE

This gadget causes the Tri-View to be cleared, and the global scene for the Take to be loaded.

DELETE ALL IMAGES

This gadget deletes all the images associated with the Take. You should use this at any time that you have modified the Take in any that would make the rendered images invalid.

PREVIEW

This gadget starts a moving wire frame preview. The preview will be repeated when it is finished. The preview may be stopped at any time by pressing the right mouse button or the escape key. The speed can be controlled with the keys 0 to 9.

RENDER ALL

This gadget initiates the rendering process. When all the the images have been produced, if a RAM animation mode has been specified, an animation file is saved to disk.

OK

This gadget permits you to leave the requester.

Current frame:

This gadget is only available when the 'Take mode' is set to FRAME. You may use this gadget to specify which frame number you are interested in. Key frame: This gadget can be toggled between YES and NO. It is used to specify whether the current frame is a key frame.

Duration:

This gadget can be used to specify the length of time, in jiffies, that the current frame should be displayed.

Image drawer:

This gadget can be used to specify a drawer for the current image. This gadget is particularly useful if you are

FUNCTIONAL REFERENCE

using floppy disks and all the images will not fit on a single disk. If this item is left blank, and an image drawer is specified for an earlier frame, then that drawer will be used for the current frame.

Scene drawer:

This gadget can be used to specify a drawer for the key frame scene for the current frame. This gadget is particularly useful if you are using floppy disks and all the scenes will not fit on a single disk. If this item is left blank, and a scene drawer is specified for an earlier frame, then that drawer will be used for the current frame.

SAVE KEY FRAME SCENE

The objects that are in the Tri-View are saved to become the key frame scene for this frame. Any objects that are derived from the global scene will not be saved.

LOAD KEY FRAME SCENE

The key frame scene will be loaded if this frame is a key frame. Objects from the global scene will also be loaded, but these will not be able to be modified. If the current frame is not a key frame, then objects will be loaded by in-betweening, but they will not be able to be modified, unless you specifically request it.

DELETE CURRENT IMAGE

The image associated with the current frame will be deleted.

RENDER FRAME

An image corresponding to the current frame will be rendered.

APPENDIX B

GLOSSARY

ATTACH

A mechanism for describing moving objects is to attach them to a path. The attachment is performed by making the object an offspring of a motion path in the hierarchy of names.

AXES

The plural of axis.

AXIS

A principal direction that is used for measuring the position of an object. In Sculpt Animate, the axes are directed to the east, north and upwards.

CENTER

The Center of a group of vertices is defined as the center of the smallest rectangular box that would enclose the vertices. The box is aligned with the coordinate axes.

CENTROID

The centroid of a set of vertices is the average of their individual positions. If the vertices all had the same mass, it would be their center of gravity.

CLICK

The expression 'click on a gadget', means that you are to point to the gadget and press down the left mouse button and then release it. If you press the button, and then decide that you want to change your mind, some gadgets permit the command to be cancelled by moving the pointer away from the gadget before releasing the button.

COMPOSITE ENCODER

A piece of hardware that takes a signal containing separate signals for the intensity of the image in red, green and blue (RGB) and combines them into a single signal. Many video recorders can only accept signals in composite format.

GLOSSARY

CURSOR

This is an object that represents a position in three dimensions, so it is displayed in all three Tri-View windows. It can be moved by the mouse, when the left button is pressed down.

CURVE

In Sculpt Animate, a curve is a series of vertices linked by edges. Each vertex on a curve must have either one or two edges associated with it.

CUSP

A point where two curves meet with a different slope. A spline may have any number of cusps. Place them at points where you wish the curve to have a corner or kink.

DELTA DATA

A description of the difference between two images. Given one image and the appropriate delta data, then another image can be reconstructed. If two images are very similar, then the size of the delta data is much smaller than the size of the image.

DOUBLE CLICK

Because the mouse only has two buttons, you can signal the computer that you wish to do something special, by pressing a button twice in rapid succession. This is called double clicking, and it is almost always done with the left mouse button. You can use the Workbench Preferences tool to set the interval of time that the Amiga regards as 'rapid succession'. See the book "Introduction to Amiga" for more details. With Sculpt Animate you should set the interval to be fairly short, so that a double click is a very deliberate act, rather than normal frenzied mouse activity.

DRAG BAR

This is a gadget that is located in the top border of each window of the Tri-View. It is used to move the window around on the screen.

GLOSSARY

EDGE

In Sculpt Animate an edge is a line joining two vertices. An edge is visible as a dark line in the Tri-View, but is never seen in a picture.

EDGE BUILDING GADGET

This gadget resides in the lower left border of each Tri-View window. Its icon is a triangle. If exactly two vertices are selected, then an edge will be constructed between the vertices. If three vertices are selected, then they will be joined by three new edges.

FACE

In Sculpt Animate objects are built from a patchwork of triangular faces. A face has a particular color and surface texture. A face is automatically created whenever three vertices are joined by three edges.

FRAME

One image of an animation. Often a frame is made up by interlacing two fields, one containing odd numbered scan lines and the other containing even numbered scan lines.

FRAME BUFFER

A piece of hardware that is controlled by the computer that accepts data for each pixel and generates a video display or a video signal. To be useful, a frame buffer should produce an image of higher resolution or with more colors than the computer's graphics hardware can produce.

GADGETS

These are small images that can be used to perform an operation. Some gadgets are common to most Amiga programs while others are unique to Sculpt Animate. The Tri-View has a number of gadgets in the borders of each window, that are used for changing the position and size of the Tri-View and for manipulating objects. Other gadgets appear in requesters. Most gadgets are operated by clicking on them with the left mouse button. Others, called sliders, are like the controls on a hi-fi graphic equalizer. You can click on the control knob and then slide it up and down, or from side to side, in order to adjust a value. Finally there are

GLOSSARY

gadgets, that when selected by clicking, will enable you enter a name or a number from the keyboard.

GLOBAL ANIMATION

An animation technique where a single scene contains all the information about objects and their motion, so that many frames can be generated from the single scene.

HELICES

The plural of Helix.

HELIX

A curve that is produced by sweeping a point around an axis and changing the distance along the axis. In Sculpt Animate, the distance from the axis can also change, resulting in a spiral curve.

HIERARCHY

A system for organizing elements into levels, where an element is never linked to another element at the same level, is never linked to more than one element at the next highest level, but may be linked to any number of elements at the next lowest level. In Sculpt Animate the elements are names that may be given to groups of vertices, paths, lamps, etc.

The part of the sky that appears to be at ground level.

IN-BETWEENING

The process of interpolation, whereby the position and shape of an object at one time is inferred from its position and shape at other times. The frames for which the scene is explicitly provided are called key frames.

INDICATED

Marking an object by placing the cursor on or near it in the current window of the Tri-View.

GLOSSARY

INDICATED VERTEX

If a vertex lies within a few pixels of the cursor in the active window of the Tri-View, this vertex is considered the indicated vertex. If more than one vertex meets this criteria and they are distributed in depth, then the indicated vertex is the one that is closest to the cursor in the depth direction.

JIFFY

The length of time that it takes to display a single field of an image. On NTSC systems this time is 1/60th of a second. On PAL systems the time is 1/50th of a second.

KEY FRAME

A frame at which all details of the scene are explicitly provided. The scenes that correspond to not key frames must be interpolated from key frames before and after the non-key frame. This process is called in-betweening.

KNOT

A point on a spline that is used to define the shape of a spline. If a knot is moved, nearby non-knot vertices will also move.

LOCAL ORIGIN

The local origin of a named object is the point that can be attached to a path. A local origin can either be fixed in space, or locked to a vertex.

LOCATION

The point from which a scene is viewed in order to render an image.

LOOP

In Sculpt Animate, a loop is a series of vertices linked by edges where each vertex is associated with two edges. It is a closed curve.

OBJECT CYCLING

A method of producing an animation where a single scene contains many objects, each representing one stage of a motion. When the scene is rendered for a particular frame, only one object is used.

GLOSSARY

OBSERVER

This entity embodies all the attributes required to specify the process of generating an image, including a viewing location, a target position, an imaging method and an image size. The width of view can be controlled by choosing a lens.

OFFSPRING

In a hierarchy, an offspring is an element at the next lowest level. In Sculpt Animate, the maximum number of direct offspring is limited to 255.

ORIGIN

The initial position of the cursor. If the coordinate window is displayed, the position of the cursor is specified relative to an origin that you can set.

PARENT

In a hierarchy, a parent is the element at the next highest level. An element may not have more than one parent.

PATH

A description of the motion of a moving object. In Sculpt Animate, a path is represented by a chain of vertices linked by edges. Each vertex represents the location of the object at a time corresponding to a particular frame.

PIXEL

A picture element on the screen. The Amiga can display 320 or 640 pixels in the horizontal direction and 200 or 400 pixels in the vertical direction. Subject to some restrictions, each pixel can be displayed in one of 4096 different colors and intensities.

POINTER

This is the image that moves on the screen when you move the mouse. It looks like a red arrow, unless you or someone else has changed it with the Preferences tool.

PREVIEW

The process of rapidly generating a moving image of an animation. In Sculpt Animate, the preview is displayed in wire frame mode.

RAM ANIMATION

A method for displaying an animation using a computer. Each image is stored in memory in a compressed form and is expanded before being displayed.

RENDER

The word render is used to apply to the process that generates an image of a scene.

REQUESTER

This is an object that appears on the screen from time to time, when Sculpt Animate needs some extra information, or wants to be sure that you have received a message before continuing. Requesters contain gadgets that may be like push button, or sliders or even little boxes where you can type a short message. Whenever a requester pops up, you must acknowledge it by answering any questions and pressing the OK or CANCEL button with the left hand mouse button before continuing with Sculpt Animate.

RUBBER BAND LINE

A dotted line drawn from a fixed point to the cursor to show where an edge could be built.

SCENE

A scene is one of the file formats used by Sculpt Animate. It can contain objects, lamps, an observer and a world. In order to be recognized as a scene, the file name should end with '.scene'. The command PROJECT SAVE SCENE automatically adds this suffix.

SCRIPT

A file that contains commands that control the program directly, rather than through mouse interaction.

GLOSSARY

SINGLE FRAME CONTROLLER

A piece of hardware that permits a computer to control a video recorder in such a way that one image is written to tape at a time.

SLOPE

The angle of a line that is parallel to a curve at a particular point. The slope at a knot helps to determine the shape of a spline.

SNAP

If the left mouse button is double clicked in a window, Sculpt Animate looks for a nearby vertex (within about 3 pixels). If more than one pixel satisfies the criteria, then the one that is closest in the 'depth direction' of the window is chosen. The selection state of the vertex is toggled. Also the cursor is moved to the position of the vertex. This is called snap. Commands are provided to snap the cursor to a vertex or the grid. Vertices can be snapped to the grid, to a spherical shape or to a plane.

SOLID OBJECT MODELING

In this manual, solid object modeling is defined as the creation and manipulation of three dimensional mathematical facsimiles of objects, for the purpose of creating images. The term has also been used to describe a different, more sophisticated process that might better be called solid object simulation, in which objects are given physical attributes such as mass and rigidity. Solid object simulation is beyond Sculpt Animate's scope, although Sculpt Animate may be used in conjunction with simulation programs.

SPEED PARAMETER

A parameter that helps to determine the shape of a spline. If a knot has a large speed parameter, the slope has an influence over the shape of a large portion of the spline.

SPLINE

A smooth curve that is generated mathematically. In Sculpt Animate, the exact shape of a spline can be modified interactively.

GLOSSARY

SWATCH

The patch of color at the lower center of one of the color requesters is called a swatch. It indicates the current color. Move the sliders to change the color.

TAKE

A sequence of frames that make up an animation. In Sculpt Animate, a Take is a file that contains enough information that the image for each frame can be rendered.

TARGET

A point within a scene that will appear in the center of a rendered image.

TOGGLE

When a gadget has two different states that it can be in, and a single action causes it to change from one to another, it is said to toggle. For example some requesters have a box that may either contain the word YES or the word NO. Clicking on the gadget will change the gadget from YES to NO or vice versa.

TOOL

In Sculpt Animate, a tool is associated with the cursor. The default tool, i.e. the one associated with the cross-hair cursor is used for indicating things and for creating vertices. The grabber gadget changed the default tool into the grabber tool which is used for moving selected objects. Other tools include the selector and deselector tools, the magnet tools for deforming objects, the curve tool for drawing freehand curves, the extrude tool for manipulating extrusions and the edge maker tool for making edges.

TRI-VIEW

The set of three windows that are used for manipulating three dimensional objects. The Tri-View represents a block of three dimensional space, and only objects within that block are displayed. Objects are displayed as wire frame images.

GLOSSARY

TUMBLE AXES

A set of three directions that define the orientation of an object as it moves along a path. By rotating the different tumble axes along a path, an object can be made to spin when it is animated.

VERTEX

This is a point in space that can define the end of an edge, or the corner of a face. In Sculpt Animate, you manipulate an object by changing the position of one or more vertices.

VERTICES

The plural of vertex.

WINDOW

A portion of the screen that sometimes can be moved around or rearranged in depth. The most important windows in Sculpt Animate are the three windows of the Tri-View. These windows have a blue and white border area containing a myriad of gadgets. The interior of the Tri-View windows are brown and contain a wire frame rendering of the contents of the Tri-View.

ZENITH

The point in the sky that is directly overhead.

APPENDIX C

SCULPT ANIMATE FILE FORMATS

The files generated by Sculpt Animate follow the IFF standard, with the exception of the RGB files described in chapter 3. The IFF format is described in the Amiga ROM Kernel Manual, and sample programs to read and write such files may be found in most Amiga public domain disk libraries. Some programming experience is helpful in understanding the format, in particular, the C language will be used to describe the nonstandard data chunks.

The file formats used by the Sculpt series of programs for the Amiga are constantly changing, through the addition of new chunk types and the definition of reserved 'padding' bytes in existing chunks. In most cases this will not present problems either with older versions of the program, or with future versions. Care should be taken to initialize unused padding bytes to zero, to maintain future compatibility.

Image files

Images are written as ILBM IFF files and need little further explanation here, since they conform to the published standards. Because of the variety of image size and mode that Sculpt Animate uses, programs which intend to use these images shouldn't make assumptions about these items. Image files have names that end with '.image'.

Images saved in conjunction with frame buffer usage are non-standard, but follow the ILBM format. These images have no CMAP chunk, so the bit planes represent actual 24-bit color intensity values: 8 planes each of red, green, and blue--stored in that order. Within each color, the most significant bits come before less significant bits.

FILE FORMATS

Scene files

Scene files contain a number of data chunks. Each chunk is described below in terms of a C structure. The comments beside each structure member should make its meaning clear. The FORM name for a scene file is 'SC3D'.

```
/*   Chunk 'LAMP' contains one or more of the following structures,
    the number of structures is determined by the length of the chunk
*/
```

```
struct elamp
{   long int pos[3];           /* The position of the lamp */
    long int brightness;      /* The brightness of the lamp */
    unsigned char color[3];    /* The color of the lamp as a triple
                                of RGB values, range 0 to 255 */

    char pad; /* Unused */
};
```

```
/*   Chunk 'OBSV' contains a specification of an observer according
    to the following structure
*/
```

```
struct observer
{   long int obsmode; /* Rendering mode as follows
                        0 Painting
                        1 Snapshot
                        2 Photo
                        3 Wireframe
                        4 Sketch
                        5 Scanline painting
                        6 Scanline snapshot
                        */
    long int fl; /* Focal length of lens
                  in millimeters
                  */
    long int althresh; /* For internal use only */
    long int threshold; /* For internal use only */
    long int robsp[3]; /* Position of observer */
    long int rtarget[3]; /* Position of target */
    short int hires; /* 0 for low and 1 for high resolution */
    short int lace; /* 0 for non-interlace 1 for interlace */
};
```



```

short int lens;      /* Lens type as follows:
                      0 Normal 50mm
                      1 Wide angle 28mm
                      2 Telephoto 135mm
                      3 Special, as given by spfl
short int manexpflg; /* 0 for auto exposure, else manual */
long int spfl;       /* Focal length of special lens */
long int expoverride; /* Current value for exposure override */
long int manexpval;  /* Value when in manual mode */
long int picsize;    /* Image size as follows
                      0 Tiny
                      1 Small
                      2 Medium
                      3 Full
                      4 Jumbo
                      5 Video
                      */
long int tilt;       /* Tilt angle */
long int aamode;     /* Anti aliasing mode as follows:
                      0 None
                      1 Good
                      2 Best
                      */
short int dithatten; /* Dithering attenuation as follows:
                      0 Standard
                      50 Half standard, etc
                      100 None
                      */
short int colorlock; /* Non zero for locking colors */
short int explock;   /* Non zero for locking exposure value */
short int expexp;    /* Internal use only */
long int expmant;    /* Internal use only */
unsigned char wfcoll[4],wfcoll2[4];
                      /* Colors for wire frame images */
short int displayearly;
                      /* Non zero to request early display
                      of rendered image */
short int dummy[29]; /* Reserved for future use, should be set
                      to zero.
                      */
};

```


FILE FORMATS

/* Chunk 'WRLD' contains a specification of the world using the following structure
*/

struct world

{long int groundmode; /* 0 None, 1 solid, 2 checkered */
long int skymode; /* 0 None, 1 solid, 2 graduated */
long int checkscale; /* Size of checkered ground */
unsigned char backbright[3];

/* Intensity of background illumination
as RGB values on a scale of 0 to
255

*/

unsigned char grcol1[3],grcol2[3];

/* Colors of ground squares as RGB
values on a scale of 0 to 255

*/

unsigned char skycol1[3],skycol2[3];

/* Sky colors as RGB values on
a scale of 0 to 255

*/

long int dummy[20]; /* Reserved for future use, should be set
to zero

*/

};

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

17/11/2023

FILE FORMATS

```
/*   Chunk 'VERT' contains an array of the following structure, one
    element for each vertex to be included in the scene
*/
```

```
struct evertex
    {long int pos[3];    /* Position of vertex */
    };
```

```
/*   Chunk 'EDGE' contains an array of the following structure, one
    element for each edge in the scene
*/
```

```
struct eedge
    {long int evertexi[2]; /* Indices denoting vertices in VERT
                                chunk
                                */
    };
```

```
/*   Chunk 'FACE' contains an array of the following structures, one
    element for each face in the scene
*/
```

```
struct eface
    {long int evertexi[3]; /* Indices denoting vertices in VERT
                                chunk
                                */
    unsigned char color[3];
        /* Face color as RGB values in the range
           0 to 255
           */
    unsigned char texture; /* The most significant bit is set
                                if smoothing is needed. The remaining
                                bits represent a texture value as follows:
                                0 Dull
                                1 Shiny
                                2 Mirror
                                3 Luminous
                                4 Glass
                                5 Metal
                                */
    };
```

1. Introduction
2. Background
3. Methodology
4. Results
5. Discussion
6. Conclusion
7. References
8. Appendix
9. Glossary
10. Index

FILE FORMATS

/* The chunk 'HIER' contains an array of the following structures,
one element for each name in the hierarchy

*/

struct ehier

{short int parindex; /* index to the hierarchy element
that is the parent of this element

*/

char name[10]; /* Name of hierarchy element */

short int type; /* Type of element, as follows:

0 Empty

2 Lamp

4 Path

6 Target

8 Observer location

10 Vertices

Bit 0x10 is set if a local origin is
used by this element

*/

short int loverti; /* index to the vertex that is the local
of this element

*/

long int lo[3]; /* Local origin position */

};

/* Chunk 'VNAM' contains an array of the following structure, one
element for each vertex that has a hierarchy name

*/

struct ename

{short int object; /* index to vertex */

short int name; /* index to hierarchy name */

};

/* The chunk 'LNAME' contains an array of short integer values
representing the hierarchy indices for each lamp. Unnamed
lamps are given an index of -1.

*/

FILE FORMATS

/* The chunk 'PATH' contains an array of the following structure,
one element for each vertex that is a part of a path.
*/

```
struct epath
{int evertexi; /* Index to vertex */
 long int etumbleaxes[3][3];
      /* Tumble axis directions, scaled by a factor of
      1L<<30
      */
 int terminator;
      /* Code value with bits set as follows:
      1 end of path
      2 end of loop
      4 interpolated
      */
};
```

/* The chunk 'KNOT' contains an array of the following structure,
one element for each knot vertex included in the scene
*/

```
struct eknot
{long int eslopes[2][3];
      /* Knot slopes scaled by 1L<<30 */
 long int espeeds[2];
      /* Speed values scaled by 1L<<30 */
 short int evertexi;
      /* index to vertex */
 short int terminator;
      /* Code value with bits set as follows:
      1 end of spline
      2 end of loop
      4 interpolated
      8 cusp
      */
};
```

FILE FORMATS

```
/* The chunk 'SPLN' contains an array of the following structure,
one element for each non-knot spline vertex
*/
```

```
struct espline
{
    short int evertexi;
        /* Index to vertex */
    short int evertknoti;
        /* Index to knot */
};
```

Take files

A take is automatically saved to disk at frequent intervals as a file whose name ends with '.take'. It is an IFF file with a form name called 'TAKE' containing the following chunks.

```
/* Chunk 'THDR' contains the following structure:
*/
```

```
struct take
{
    int frmmode; /* set to 1 for frame mode */
    int rammode; /* set as follows:
                    0 not RAM animation
                    1 normal RAM animation
                    1 economy RAM animation
                */
    int numframes; /* number of frames in the take */
    int curframe; /* current frame number */
    int saveimages;
        /* set to 1 if frames are to be saved to disk */
    int prevsize; /* preview size as follows:
                    2 Medium
                    3 Full
                */
    int motionblur;
        /* set for motion blur */
    int loopmode; /* set as follows:
                    0 no loop
                    1 loop
                    2 oscillation
                */
};
```

FILE FORMATS

```
int framebuffer;
    /* set if frame buffer is in use */
int framecontroller;
    /* set if frame controller is in use */
char anidrawer[22];
    /* animation drawer name */
char foreground[22];
    /* foreground file name */
char background[22];
    /* background file name */
char pad1[22];
    /* unused, should be set to zero */
char fcname[22];
    /* frame controller name */
char pad2[22];
    /* unused, should be set to zero */
char fbname[22];
    /* frame buffer name */
short int colorlock;
    /* set when colors are locked */
char creg[96];
    /* current color registers */
short int explock;
    /* set when exposure values are locked */
short int expexp;
    /* internal use only */
long int expmant;
    /* internal use only */
short int obsmode;
    /* Rendering mode as follows
        0 Painting
        1 Snapshot
        2 Photo
        3 Wireframe
        4 Sketch
        5 Scanline painting
        6 Scanline snapshot
    */
short int hires;
    /* 0 for low and 1 for high resolution */
short int lace;
    /* 0 for non-interlace 1 for interlace */
```

FILE FORMATS

```
short int picsize;  
    /* Image size as follows
```

```
    0 Tiny  
    1 Small  
    2 Medium  
    3 Full  
    4 Jumbo  
    5 Video
```

```
    */
```

```
short int bitplanes
```

```
    /* number of bit planes */
```

```
unsigned char wfcoll[4],wfcoll2[4];
```

```
    /* colors for wire frame rendering */
```

```
char padding[64];
```

```
    /* unused, should be set to zero */
```

```
};
```

```
/* The remaining chunks are of type 'TFRM', one for each frame in  
the take. Each chunk contains the following structure
```

```
*/
```

```
struct etakeframe
```

```
{ short int framenum; /* Frame number */
```

```
  short int keyframe; /* Set to 1 if it is a keyframe */
```

```
  short int duration; /* Duration, in jiffies */
```

```
  char imadrawer[22]; /* image drawer name */
```

```
  char scedrawer[22]; /* scene drawer name */
```

```
  char padding[64]; /* unused, should be set to zero */
```

```
};
```

APPENDIX D:

ERROR MESSAGES

This appendix contains a list of error messages that can be encountered during the use of Sculpt Animate. Messages that simply ask for confirmation of a command, etc., have generally been omitted.

The messages are grouped topically. Messages with similar meanings are explained once, as a group.

Messages relating to memory:

Not enough memory

Not enough memory to display image

Not enough memory to render image

Not enough memory for preview

These errors indicate that there isn't enough memory to perform the requested operation. This can occur even if plenty of expansion RAM is available, since chip (graphics) ram is always limited. Note that the memory status display is only updated occasionally during renders.

Memory too fragmented

This error indicates that the largest available contiguous memory block is not large enough to carry out the operation.

Not enough memory or disk space

The requested operation may be possible if more disk space is made available, either in the program's current directory or in a mounted device or volume named "SATEMP:" (intermediate image data would be stored here). No more than 600K of additional disk space should be required in any case.

Not enough memory for task or recovery

The program has "painted itself into a corner", and can't find enough free memory to continue at all. This should be rare.

Unable to open screen

Unable to open window

A system call to open a screen or window has failed. This is most likely because of a lack of chip (graphics) memory.

ERROR MESSAGES

Messages relating to files:

File does not exist!

Cannot rename non-existent file!

Cannot delete non-existent file!

Take not found. Create new Take?

The file requested does not exist in the selected directory. In the case of a Take, clicking OK will enable the Modify Take command, which saves a Take file automatically.

Unable to open file

Unable to open image file

A file cannot be found that is required for the processing of a take. This is most likely to happen if a name is improperly entered into the foreground or background image slot, if CANCEL is clicked in an "insert volume" requester, or if a key frame has been specified for which no scene has been saved. This can also happen if the file is being used by another program which is running concurrently.

Unable to create file

Unable to open file for animation

Unable to open files for bit-planes

A system I/O error occurred which prevented a file from being opened for storing data. This can happen if a protected file of the same name exists on the requested directory, if a file of the same name is being used by another program, if the requested directory doesn't exist, if the disk is not mounted or is write protected, etc.

Unable to open device

Fail to open Frame buffer

Fail to open Frame controller

The selected device driver was either not found or not successfully loaded by the system. Make sure the driver filename matches it's device name (case sensitive), and that it is properly installed in DEVS:.

Unable to close file for bit-planes

The program was not able to access a volume on which it had been writing bit-plane data. This can happen if CANCEL is clicked on a volume requester.

Error reading from file

Error reading from animation file

A system I/O error occurred while the file was being accessed. This may mean that the file has been corrupted. This can also happen if the end-of-file is reached unexpectedly.

ERROR MESSAGES

Error writing to file

Failure while writing to disk

Error writing animation file

Disk error on bit-planes file

A system I/O error occurred while data was being stored. This is most likely to happen if the volume being written to fills up. Try specifying a larger volume in the "Animation drawer" slot of the Modify Take requester, Global mode. Or, in the Frame mode, name different volumes in the "image drawer" slot for various frames.

This file does not contain a scene

No image found on this file

This file does not contain a Take

The supposed Scene, Image, or Take file that is being opened does not actually contain valid data of that type.

The file is corrupted

The scene file contains 'garbled' data that can't be used.

The image has no color information

The image has no header information

The image file does not contain specific data which is required to properly display the image.

This image file does not match frame buffer

An image that was supposedly saved as part of a frame buffer render is the wrong size/resolution for the current frame buffer. This can happen if you change the frame buffer's settings or select a new frame buffer when images already exist for a Take. Click DELETE ALL IMAGES when you change settings.

Fatal disk read error

Fatal disk write error

A serious system I/O error has occurred which makes it impossible for program execution to continue.

Messages relating to hierarchies:

Hierarchy error. It may have been truncated.

The scene's hierarchy is corrupted. This may have occurred as a result of a system I/O error, garbled file, etc.

No indicated vertex

"Connected" commands require that the cursor be "indicating" a vertex, i.e., placed within a pixel or so in the selected window.

ERROR MESSAGES

A lamp cannot be given this name

A path cannot be given this name

Vertices cannot be given this name

This tells you that the word "location" or "target" was the current name in the hierarchy when you clicked OK. If you just wanted to name the observer's location or target, then there is no problem. Otherwise, use another name.

Messages relating to rendering:

Observer mode needs to be set

The loaded scene requires a rendering mode that has not been implemented in this version. A different mode must be set.

Already making a picture

A START or RENDER command has been issued while an image was already being rendered. Wait for the render to finish, or use Abort.

ERROR: images have different size or type

ERROR: colors not locked

Delete all images and render them again

The images that have been saved in a take must all have the same size, resolution, type, and color map in order to be compressed for RAM playback. These parameters are stored in the ".take" file. They may be wrong if you don't DELETE . ALL IMAGES before beginning a new render, or if the Take file is somehow lost or altered between rendering sessions.

Messages relating to Takes:

...and erase the Take?

The command ERASE ALL was called after a take had been loaded. If you click "OK" to this query, the take will be erased from RAM. If you have made any settings, they will remain on disk in a '.take' file.

Frame number too small

A "current frame" number of zero (0) was entered. Frames are counted from one (1) in Sculpt Animate.

More frames needed

A PREVIEW requires a take of at least two (2) frames.

ERROR MESSAGES

Not a Key Frame! Generate scene by in-betweening?

The selected frame is not a key frame, so (presumably) no key frame scene has been saved for it. If you click OK, the program will interpolate a new scene from neighboring key frame scenes.

Keyframes n1 and n2 have different structure

The keyframes that bracket the frame being rendered (or loaded) were constructed differently. See the "in-betweening" section of chapter 11 for details.

Messages relating to Paths:

Not a path

The desired operation can only be performed on a path. The indicated vertex is not a part of any path.

No path indicated

The command will not work unless a vertex that is part of a path is indicated (has the cursor placed on it in the current window).

Already is a path

An attempt was made to make a path from vertices that were a part of another path.

Vertex has no edge

Vertex has more than 2 edges

The indicated curve does not describe an unambiguous course for a path to follow.

Path direction not shown

Exactly one vertex must be selected adjacent to the indicated vertex when making a closed loop path, to specify the path's direction of travel.

Path is not a loop

If a curve is not a closed loop, an end vertex of the curve must be indicated for it to be made a path.

You are already at the end of the path

The NEXT and PREVIOUS gadgets in the Tumble Axes requester will only work if there is a next or previous vertex (respectively).

ERROR MESSAGES

Path too short

SHOW PATH POSITION was used while the current frame number in the Modify Take requester was larger than the path's number of vertices.

Messages relating to the Tri-View:

Tri-View cannot be moved any further in this direction

Tri-View cannot be made larger

Tri-View cannot be made smaller

The largest allowable coordinate in Sculpt space is plus or minus 10,000,000 fundamental units (no decimals). Tri-View size and motion are limited to avoid going too far past this boundary. The Tri-View also cannot be made smaller than a size of about 1 unit = 1 pixel. Some space is reserved beyond these limits to ensure computational accuracy.

Miscellaneous editing messages:

All selected vertices must lie on a plane

The desired operation only works on vertices that all lie within the same plane.

At least 4 vertices must be selected

The FILL command works on loops of four or more vertices.

Bad control curve

The indicated curve is not adequate for the program to use as a guide in constructing a helix.

Error: part of another spline

An attempt was made to create a spline using vertices that are already part of another spline. This is not allowed.

Error: selected vertices are not connected

An attempt was made to create a spline using vertices that are not connected.

Loop does not lie in a plane

At least one of the loops selected for an UNSLICE does not lie in a plane.

Loop planes are not parallel

The loops selected for an UNSLICE do not lie in parallel planes.

ERROR MESSAGES

Must be greater than 3!

An attempt was made to create a helix with 3 or fewer rotational steps per cycle. This is not allowed.

No lamp near the cursor

The grabber was invoked with no vertices selected, and no lamp within grasp.

No unique simple selected curve

There is no helix control curve indicated, and the selected vertices (if any) are not adequate for building a helical surface.

No vertices selected

The selected command operates only on selected vertices.

Not a spline

The selected command operates only on splines.

Not a valid curve for a spline

A curve which is to be a spline must be a simple, fully-connected curve with no more than two edges connected to any vertex.

Nothing to fill!

The selected vertices are already filled.

Number too big, try again

A number was entered into a text entry gadget that is too large for the given application (e.g., exposure value).

How many vertices are to be inserted?

SUBDIVIDE was called with two unconnected vertices selected. You may specify a number of vertices to be inserted between these two, creating a curve. (Good for paths.)

Select the object to be expanded

The EXPAND command was called without selecting any vertices.

This vertex is not a knot

A knot of a spline must be indicated by the cursor before Modify Knot will work.

ERROR MESSAGES

Script error messages:

It is possible that errors can occur during the execution of a script file or keyboard macro, as a result of typos or incorrect entries in the script. When this occurs, this message will appear:

Script syntax error nn on line nnnn. Continue?

The "line" number tells which line of the script is at fault. It is possible that the error actually occurred on the preceding line (as when a parameter has been omitted). If you click OK, script execution will continue with the next line.

The syntax error number can provide a clue about what has actually gone wrong. The following is a list of script syntax errors that can appear.

- 1 A name enclosed in double quotes was not found.
- 2 The first word of a command is incorrect.
- 3 A command that moves the cursor is incorrect.
- 4 When saving a scene, something other than one of the following keywords was found: SElected, LAMps, OBServer, WORld or OBJects.
- 5 When loading a scene, something other than one of the following keywords was found: SElected, LAMps, OBServer, WORld or OBJects.
- 6 The number representing an expansion factor was entered incorrectly.
- 7 A number representing a color value has been incorrectly entered.
- 8 The number representing the size of a sphere was not supplied, or supplied incorrectly.
- 9 The number representing the size of a hemisphere was not supplied, or supplied incorrectly.
- 10 The number representing the size of a cube was not supplied, or supplied incorrectly.
- 11 The number representing the size of a prism was not supplied, or supplied incorrectly.
- 12 The number representing the size of a disk was not supplied, or supplied incorrectly.
- 13 The number representing the size of a circle was not supplied, or supplied incorrectly.

ERROR MESSAGES

- 14 The number representing the size of a cylinder was not supplied, or supplied incorrectly.
- 15 The number representing the size of a tube was not supplied, or supplied incorrectly.
- 16 The number representing the size of a cone was not supplied, or supplied incorrectly.
- 18 The integer representing the number of decimal places to use in coordinates was not supplied correctly.
- 19 The number representing the focal length of a lens was not given correctly.
- 20 The integer representing the exposure override percentage was not correctly supplied.
- 21 The observer tilt angle has not been correctly supplied.
- 22 The size of the ground checkerboard has been incorrectly specified.
- 23 The angle of rotation for a rotate command has not been given correctly.
- 24 The integer value specifying the number of bitplanes has been incorrectly specified.
- 25 The keywords SElect or DESelect were expected, but not found.
- 26 An invalid number was used to represent a color.
- 27 The number representing the grid size was not given correctly.
- 28 The cursor was not positioned on a vertex in order to indicate a knot.
- 29 The indicated vertex is not a knot.
- 30 A name enclosed in double quotes contains a space, return or non-printing character.
- 31 A name enclosed in double quotes is not terminated, or is too long.
- 33 The second part of a command is incorrect.
- 34 An integer representing the number of vertices on the perimeter of a disk was not correctly supplied.
- 35 An integer was expected, but not found.
- 36 An integer representing the number of subdivisions of a sphere was not correctly supplied.
- 37 An integer representing the number of subdivisions of a hemisphere was not correctly supplied.
- 38 An integer representing the number of vertices on the perimeter of a circle was not correctly supplied.

ERROR MESSAGES

- 39 An integer representing the number of vertices on the perimeter of one end of a cylinder was not correctly supplied.
- 40 An integer representing the number of vertices on the perimeter of a tube was not correctly supplied.
- 41 An integer representing the number of vertices on the perimeter of a cone was not correctly supplied.
- 42 The number representing the percentage of standard dithering has been incorrectly supplied.
- 43 An error was made while specifying a direction.
- 44 The number specifying the speed at a knot was not given correctly.
- 45 The cursor is not indicating a vertex for a tumble.
- 46 The indicated vertex does not lie on a path.
- 47 An error has been made while specifying the first direction for a tumble axis.
- 48 An error has been made while specifying the second direction for a tumble axis.
- 49 The two directions to specify a tumble axis are parallel.
- 50 No take loaded.
- 51 Bad value for the number of frames for a take.
- 52 Invalid number for current frame.
- 53 Invalid number for frame duration.
- 54 Bad values for Tri-View sizes.
- 55 Hierarchy name error.
- 56 Macros can only be assigned to printable characters.
- 57 Macro too long, must be less than 30000 bytes.

INDEX

- ABORT 3-17, A-30
- ABOUT A-10
- absolute origin 10-14, 13-4
- ADD
 - CIRCLE 5-4, A-19
 - CONE 5-4, A-20
 - CUBE 5-4, A-19
 - CYLINDER 5-4, A-19
 - DISK 5-4, A-19
 - DUPLICATE A-18
 - EDGES 4-3, A-20
 - HEMISPHERE 5-4, A-19
 - LAMP 4-4, 7-1, A-20
 - PRISM 5-6, A-19
 - SPHERE 5-1, A-18
 - TUBE 5-4, A-19
 - VERTEX 4-2, A-20
- ALT key 2-16, 4-13, 6-4, A-35
- Amiga key 2-16, 6-4
- angles 13-6
- ANIM format 11-22
- animated object 12-15
- animations 1-5
- anti-aliasing 3-11, A-29
- attach 1-6, 10-11, B-1
- AXIS B-1
- back gadget 2-4, 2-21, A-2
- background 7-5
- background image 12-16, A-40
- backup 2-1
- BATCH 14-20, A-10
- BIT PLANES 3-6, A-27
- CANCEL 2-10, 9-8, 11-8
- cardinal direction 2-2
- CENTER 6-7, B-1
- Center Tri-View Gadget 2-16, A-3
- centroid 6-7, B-1
- CHECKERED A-32
- chip memory 3-17, 3-18
- CLICK B-1
- color, faces 4-10, A-35
- color fringe 3-3
- color lock 12-9
- color map 3-3
- Commodore key 2-16
- composite encoder 12-20, B-1
- Contract Tri-View Gadget 2-17, A-3
- COORDINATES 13-2, A-22
- COPY 9-10, A-34
- current name 9-5
- CURSOR B-2
- cursor 2-5, 2-16
- CURVE 5-6, A-24, B-2
- cuspid 8-7, A-36, B-2
- CUT 9-10, A-34
- DELETE ALL IMAGES 11-25, A-41
- DELETE CURRENT IMAGE 11-25, A-42
- DELETE HIGHLIGHTED FILE 2-9
- DELTA DATA B-2
- DESELECT
 - ALL 4-7, A-11
 - CONNECTED 4-7, A-12
 - INDICATED EDGE 4-8, A-12
 - INDICATED VERTEX 4-8, A-12
 - NAMED VERTICES A-12
 - SWAP A-12
- DESELECTOR 6-11, A-23
- disk space 12-5
- DISPLAY EARLY 3-15, A-30
- DISPLAY LATE 3-15, A-30
- dither 7-6

INDEX

- DITHERING 3-12, A-30
- DO FILL 5-7, A-16
- DO HIDE SELECTED
VERTICES A-17
- DO MAKE HELIX A-16
- DO MAKE PATH 11-7, A-17
- DO MAKE SPLINE A-17
- DO MAKE TRI-VIEW BIG
A-18
- DO MAKE TRI-VIEW
SMALL A-18
- DO REFLECT A-16
- DO REVEAL HIDDEN
VERTICES A-18
- DO SHOW PATH
POSITION A-17
- DO SPIN 5-9, A-15
- DO SUBDIVIDE A-15
- DO UNSLICE A-16
- DOUBLE CLICK B-2
- Double buffering 11-26
- Drag Bar 2-3, A-2, B-2
- drawer 2-9
- drive 2-8
- DULL 4-14
- DUPLICATE 6-11
- Duration A-41
- ECONOMY 11-26, A-38
- EDGE B-3
- EDGE BUILDING GADGET
A-4, B-3
- EDGE MAKER 5-8, A-25
- Edge Building Gadget 4-2,
4-5
- edge 1-2, A-1
- edge count 3-19
- edges 4-2
- EDIT ADD LAMP 2-19
- EDIT ADD sizing 5-5
- ERASE ALL 4-2, 4-10, A-14
- ERASE ALL LAMPS 4-10,
A-13
- ERASE INDICATED
EDGE 4-10, A-13
LAMP 4-10, A-13
PATH 10-3, A-13
SPLINE A-13
VERTEX 4-10, A-13
- ERASE NAMED VERTICES
A-13
- ERASE SELECTED
EDGES 4-9, A-12
VERTICES 4-9, A-12
- escape key 14-2
- EXPAND 6-5, 8-9, A-34
- Expand Tri-View Gadget
2-15, A-3
- EXPOSURE 3-11, 7-3, A-28
- exposure lock 12-9
- exposure, animation 12-10
- Extrude 1-3, 5-14, 6-12,
A-24
- FACE COLOR 4-11
- face 1-1, 4-3, A-1, B-3
- face count 3-19
- fast memory 3-17, 3-18
- FETCH COLOR 4-12
- file formats C-1
- file names 12-3
- File Requester 2-8, 2-20,
A-32
- FILL 5-7, 8-9
- film strip gadget 11-4, 11-19
- finding things 2-17
- floppy disk 2-8
- font 14-3, A-7
- foreground image 12-16,
A-40
- frame 10-2, B-3
- frame buffer 12-19, A-39,
B-3
- frame controller A-39
- frame rate 11-9
- front gadget 2-4, A-2

- gadgets 2-2, 2-5, B-3
- gearwheel 8-9
- GenLock
 - as NTSC encoder 12-20
 - with images 12-21
 - with Tri-View 12-22
- GLASS 4-14
- global animation 10-1, 11-5, B-4
- Grabber 4-17, A-25
- Grabber Gadget 4-17, 5-5, 8-3, A-4
- grid 13-7
- GRID OFF A-22
- GRID ON A-22
- grinding gears 2-8, 3-13
- ground 1-3, 7-6
- HAM 3-3
- hard disk 2-8
- HELICES B-4
- helix 5-12, B-4
- HI-RES A-26
- hidden faces 4-6
- HIDE SELECTED
 - VERTICES 6-15
- hide image 2-10
- hierarchical motion 10-12
- hierarchy 9-1, A-33, B-4
- hierarchy rules 9-6
- Hold And Modify 3-3
- hue 4-11
- icons 2-21
- IFF 2-21 C-1
- ILBM 2-21 C-1
- illumination 7-4, A-32
- image drawer 12-7, A-41
- image size 2-20, 3-4, 3-13, A-28
- image, hiding 2-20
- in-betweening 11-14, B-4
- in-betweening distortion
 - 11-18
- INDICATED B-4
- indicated vertex 4-5, B-5
- INTERLACE A-27
- Jaggies 3-11
- JIFFY B-5
- key frame 1-5, B-5
- key frame animation 11-5, 11-12
- key frame scenes 11-12
- keyboard equivalents 14-1
- knot 8-1, A-36, B-5
- lamp(s) 1-4, 2-19, 4-10, 4-18, 7-1
- lamp brightness 7-1
- lamp color 7-1
- lamps, arrangement 7-3
- lathe (DO SPIN) 5-9
- LENS 3-9, A-27
- lens, animation 12-10
- LO-RES A-26
- LOAD
 - ANIMATION 11-27, A-7
 - CODE 3-17, A-8
 - FONT 14-3, A-7
 - IMAGE 2-8, A-6
 - NAMED OBJECT 9-12, A-7
 - OBJECT 6-3, 8-10
 - SCENE 2-11, 6-2, A-6
 - TAKE 11-2, A-7
 - WORKBENCH 3-18, A-8
- LOAD GLOBAL SCENE A-40
- LOAD KEY FRAME SCENE
 - 11-13, A-42
- local origin 10-14, B-5
- location 2-18, 2-19, 3-8, 9-7, 12-10, A-27, B-5
- LOCK (local origin) 10-15
- Loop mode 12-1, B-5
- lozenge 5-5
- LUMINOUS 4-14
- MACRO 14-20
- MAGNET 6-8, A-23
- MAKE HELIX 5-10

INDEX

MAKE PATH 10-3
MAKE SPLINE 8-2
MAKE TRI-VIEW BIG 2-4
MAKE TRI-VIEW SMALL
2-4
manual organization ix
memory 3-17
menu system 2-5
METAL 4-14
MIRROR 4-14
MODES 3-1, A-25
MODIFY FACES 4-10, 5-2,
A-14, A-35
MODIFY
INDICATED KNOT 8-4,
A-14, A-36
INDICATED TUMBLE 10-5,
A-15, A-37
LAMPS 7-1, A-14, A-36
LOCAL ORIGIN 10-15,
11-7, A-14
TAKE 11-3, A-15, A-38
WIRE-FRAME COLORS 3-1,
A-14
modify knot 8-4, A-14, A-36
modify tumble 10-5, A-15,
A-37
Motion blur 12-18, A-40
mount 2-8
MOUSE A-5
Move Tri-View Gadget. 2-13,
A-2
moving lamps (in Tri-View)
4-18, 7-2
multiple commands 2-7
NAME CONNECTED
VERTICES 9-8
NAME INDICATED LAMP
12-12, A-22
NAME INDICATED PATH
10-11, A-21

NAME SELECTED
VERTICES 9-4
NTSC video
encoding 12-20, B-1
image size 3-13
'jiffy' length B-5
selecting 3-13
null name 9-7
object cycling 12-14, B-5
observer 1-3, 2-11, 2-18,
A-25, B-6
OFFSPRING 9-6, A-34, B-6
OK gadget 2-10
orbital model 10-12
orbital paths 11-6
origin 13-4, B-6
OSCILLATE 12-1, A-39
paint programs 2-21, 3-12
PAINTING 2-20, 3-2, A-25
PAL video
image size 3-13
'jiffy' length B-5
selecting 3-13
Parent 9-4, B-6
Parent Directory 2-9
PASTE 9-10, A-34
PATH B-6
path(s) 1-5, 10-1, B-6
Phong smoothing 4-15, 8-10
PHOTO 3-4, A-26
pixel(s) 3-3, 3-11, B-6
plumbing fixture 6-12
POINTER B-6
Preview 11-9, A-39, A-41,
B-7
protractor 13-6
QUIT 2-22 A-10
RAM animation 1-6, 11-23,
12-11, B-7
RAM disk 2-8
ray tracing 3-3
rectangular block 6-6

- REFLECT 6-11
- REFRESH 14-19
- RENAME hierarchy element 9-10
- RENAME HIGHLIGHTED FILE 2-9
- RENDER ALL 11-24, A-41
- RENDER B-7
- RENDER FRAME 11-12, 11-25, A-42
- REVEAL HIDDEN VERTICES 6-16
- Reverse Gadget 2-18, A-2
- RGB data 3-7
- Rotate Gadget(s) 6-4, 10-7, A-4
- rubber band line 5-7, 5-8, B-7
- saturation 4-11
- SAVE
 - GLOBAL SCENE 11-8, A-40
 - IMAGE 2-20, A-8
 - KEY FRAME SCENE A-42
 - NAMED OBJECT 9-12, A-8
 - OBJECT 6-3, A-8
 - SCENE 2-21, 6-1, A-8
- Save images (Take gadget) 12-8
- SCANLINE A-26
- SCANLINE PAINTING 3-2
- SCANLINE SNAPSHOT 3-3
- scene 2-11, B-7
- scene drawer 12-7, A-42
- scene precedence 11-5
- screen resolution 3-5
- screw thread 5-10
- script 14-6, B-7
- scroll bar, 2-10
- SELECT A-10
- SELECT
 - ALL 4-7
 - CONNECTED 4-7, 4-18, A-10
 - INDICATED EDGE 4-8, A-11
 - INDICATED PATH 10-4, A-11
 - INDICATED SPLINE A-11
 - INDICATED VERTEX 4-8, A-11
 - NAMED VERTICES 9-9, A-11
 - SWAP 4-8, A-11
- Select Gadget 4-7, A-4
- selected 4-4
- SELECTOR 6-10, A-23
- SET (local origin) 10-15
- SHINY 4-14
- SHOW ANIMATION 11-27, A-7, A-9
- SHOW IMAGE 2-20, 3-16, A-9
- SHOW PREVIEW 11-10, A-9
- side A-36
- single frame controller 12-19, B-8
- sizing gadget 2-4, 2-15, A-3
- SKETCH 3-2, A-25
- sky 1-3, 7-5, A-31
- slider 2-10
- slope 8-3, A-36, B-8
- smoothing 1-1, 4-15
- SNAP B-8

INDEX

SNAP

- CONNECTED VERTICES TO SPHERE 5-2, A-21
- CURSOR TO CENTER 6-7, A-20
- CURSOR TO CENTROID 6-7, A-20
- CURSOR TO GRID 13-8, A-20
- CURSOR TO VERTEX 13-4, A-20
- SELECTED VERTICES TO GRID 13-8, A-20
- SELECTED VERTICES TO PLANE A-21
- SNAPSHOT 3-4, A-26
- SOLID OBJECT MODELING B-8
- SOLID sky 7-6
- speed (spline knot parameter) 8-3, A-37, B-8
- spiral 5-12
- spline 8-1, B-8
- spline use 8-8
- stack 2-1
- START 2-19, 3-15, A-30
- STATUS 3-17, 3-19, A-30
- SUBDIVIDE 5-2, 5-3, 6-9
- SWATCH B-9
- Take 1-5, 11-1, A-38, B-9
- tape measure 13-5
- TARGET 2-19, 3-8, 9-7, 12-10, A-27, B-9
- text entry gadget 2-20
- TEXTURE 4-13
- TILT 3-14, A-29
- tilt, animation 12-10
- time remaining 3-19
- TOGGLE B-9
- TOOL B-9
- TOOLS CURVE 5-6
- Tri-View 1-2, 2-2, B-9
- troubleshooting 2-22
- TUMBLE AXES B-10

Tumble Axes 10-5, B-10

unit, disk 2-8

units, measure 13-3

UNLOAD

ANIMATION 11-27, A-9

CODE 3-17, A-10

FONT A-10

IMAGE 2-11, 3-16, A-9

PREVIEW 11-10, A-9

WORKBENCH 3-18, A-10

unselected 4-4

Unslice 6-14

value (color) 4-11

vertex 1-2, 4-1, A-1, B-10

vertex count 3-19

video

encoding 12-20, B-1

frame buffer 12-19, B-3

overlays 12-21

recording 12-18

VERTICES B-10

volume 2-2, 2-8, 12-7

WINDOW B-10

wire frame 1-2, 2-12, 3-1, A-25

World 6-2

ZENTH B-10

zoom 3-9

